The U.K. ATARI Computer Owners Club Issue 17 Price £1.00

Independent User Group

Monitor

Inside this Issue
The Terrorpods are coming!!
Berg - a chilling adventure
Useful Routines
for your ST

ST Reviews
GFA Draft
Fast ASM
Tempus

8-Bit Reviews
Lightspeed C
The Dungeon
Druid

XL/XE and ST

ATARI ONLY

STOCKSOFT

SUPPORTING THE EIGHT BIT

MAIL ORDER ONLY

WORLD FAMOUS HAPPY! DISC DRIVE ENHANCEMENTS

HAPPY REV 7

This is the latest enhancement for the 1050 disc drive, to come from the world famous "HAPPY" people in the U S A. It is a replacement controller board for your 1050 drive, full fitting instructions supplied, which are basically remove drive cover, unplug 2 chips, plug in Happy.

GIVES THE FOLLOWING

Makes your drive Read/Write faster
Back up including B/Sectors
True Double Density
Fast Sector copier 128k
Warpo Speed D O S
Compactor of Files
Drive Diagnostic
D I Y Kit only £169 POST FREE
1050 drive fitted with Happy only £299 post free

UPGRADE

A replacement chip turns a 1050 duplicator into a Happy, complete with Rev 7 software, will still allow you to run dupes. Rev 4 software **£45 post free**

HAPPY "OPTIONAL EXTRA"

1050 Controller **£49 post free** for use with Happy enhanced 1050 drives, only controls R/W Speeds, Write protect, switches and L E D, no soldering – full instructions.

EXTRA P.D.B FILES

for your HAPPY, a double sided disk by "DupeTec New York". Packed with new P D B Files. Also runs on "Warp Lazer" and "Hyperdrive" enhancements **£20 post free**

WARP DRIVE REV 3

UK produced replacement control board for 1050 drive, fitting as Happy.

GIVES THE FOLLOWING:

Back up including most types of bad sectors
True Double Density
PLUS True D/D Menu Maker
True D/D D O S
NOW with the ability to run Archiver Progs on the 1050 drive
PLUS P D B Files DIY Kit only
£80 post free

1050 DUPLICATOR REV 4

Based on a 6502 chip, is a replacement board for your 1050 drive. Easy 15 minute installation, compatible with both XL and XE.
Gives – read and write to B/sectors, custom formats, double sectors, also true double density, and turbo drive giving up to 18 times faster read and write.
Comes complete with all hardware, software, fitting instructions **£159**. Rush your order now! Or send large S A E. for further information.
1050 Drive with Dupe fitted only **£300 post free**
REV 4 UPGRADES for 1050 DUPE
now in stock £20 post free

SPECIAL OFFERS

The following Disk operating Systems
Only £2 each post free

(1) D O S 2
(2) D O S 2.5
(3) D O S 2.5F
(4) D O S 2.6
(5) D O S 3.00
(6) D O S 4.00
(7) UTILITY D I
(8) ART D O S
(9) MACH O O S

ABOVE ENHANCEMENTS SOLD SUBJECT TO NOT INFRINGING COPYRIGHT PROGRAMS.

FREE 24 PAGE CATALOGUE
Games tapes or disks
Utilities and enhancements
everything for the 8 bit
send large S A E

STOCKSOFT
15 WOODBROOKE ROAD, BIRMINGHAM B30 1UE

TASKMASTER

Auto-convert your tapes (single, multi, long-block, etc.) to Autoboot disk. The powerful utility includes assembler, editor, monitor functions, for both new and experienced disk drive owners.
Only £9 p & p inclusive.

RAMBIT II

Re-record your tapes (single, multi, long-block, etc.) to load around 600% FASTER with supplied cassette interface.
Kit £15 (state cassette model) **OR** fitted **£22 all inclusive.**
Both utilities are suitable for 64K 600XL, 800XL or 130XE.
Sold subject to non-infringement of copyright.
Send S.A.E. for details or PO/Cheque payable to:

RAMBIT

16 THE GREEN, THURLBY, BOURNE, LINCS. PE10 0HB.

SLUG

The ST London User Group

We were the first and we are the best! Monthly meetings, quarterly newsletter, the very best public domain software at the cheapest prices, news, reviews, tutorials, gossip, games and hardware.

Yearly subscription is £10.00 which includes four quarterly magazines, four support disks plus postage and packing, OR £5.00 for just the four magazines including postage and packing. For a sample magazine, send £1.00 only. Cheques or Postal Orders made payable to:
ST London User Group

M. Mills (SLUG), 7A Ambleside Drive, Southend-on-Sea, Essex, SS1 2UT.

Check us out!! You won't be sorry!!

THE U.K.

ATARI

COMPUTER OWNERS CLUB

INDEPENDENT USER GROUP

CONTENTS

2 **Cracking the Code**
Horizontal and vertical scrolling

9 **Eight Bit Library**
This quarters selection of new programs.

10 **Borg**
Superb adventure set in the ice cold seas of the North Atlantic

16 **8-bit Reviews**
This issue we look at Lightspeed C, Druid The Dungeon and Pirates of the Barbary Coast

19 **Scrabble Crossword**
Can you beat the computer and fill in the most words?

23 **ST Library**
All the new additions to the library are shown

24 **ST Reviews**
Includes Terrorpods GFA Draft, Fast ASM, Tempus and many more

32 **ST Routines Matter**
Part two covers getting a line of text, reading a disk file and displaymg picture files

37 **ST Programming**
This episode covers Bindings VDI calls Control and Output functions and much more

39 **Classified**
Your opportunity to sell something or find a bargain

ATARI XMAS SHOW

The Atari christmas show organised by Database is to be held at the Novetal in Hammersmith from the 20th to the 22nd of November This event has grown in stature over the last two years as THE show to go to If you are an Atari enthusiast Monitor will be there as usual and we hope to meet many of you at the show It promises to be really exciting, with lots of new software on display not to mention the Mega ST's et all! If you don t go anywhere else this year make sure you get along to this one See you there!

SUBS UP

For many years we have managed to maintain the club fee at the same level but due to increased costs we have been forced to increase the club subscription An annual membership fee will now be £5 which entitles you to four issues of Monitor and unlimited access to the PD libraries. Overseas members/subscribers should pay £8 from Europe (or surface mail worldwide) or £12 for Airmail outside Europe.

ASSOCIATED

The club has joined the Association of Atari Computer Clubs which was recently formed by Ken Ward (better known as the man that runs the Norwich User Group who publish a fine newsletter called Nugget) The aim of the association is to bring the member user groups closer together with the result of mutual benefits for all In addition, it is hoped that a better relationship with Atari Corp will be forged (and other important organisations on the Atari scene in the U.K.) so that user groups are not looked on with quite so much disdain The response to the association from the user groups has been pretty positive the response from Atari a little less so For example as this column is being typed Atari are in the throws of allocating free stand space to user groups to attend the PCW show at Olympia whilst this is to be applauded they are not doing it through the association Surely it would be better for Atari to deal with one organisation than a dozen or more. Maybe they'll get it right in time for the Atari show!

CONGRATS

The eagle eyed amongst you may already have noticed the letters that have appeared after Keith Mayhew s name in the Cracking the Code article. This is because Keith has some how managed to pass his exams at university despite spending many hours helping to get Monitor published The rest of the Editorial team (and many of you I am sure) would like to offer Keith our congratulations and wish him luck over the next 3 years as he tries for a doctorate Well done Keith!

CREDITS

Editor	Roy Smith
Art Editor	Greg Buckley
Technical Editor	Ron Levy
Technical Editor	Keith Mayhew
Adventure Editor	Steve Hillen
ST Librarian	Mike Stringer

Cover: Terrorpods copyright Roger Dean and Psygnosis

Club Address U K Atari Computer Owners Club, P O Box 3, Rayleigh, Essex SS6 8LR.

ADVERTISEMENTS

Please note that the club cannot be held legally responsible for claims made by advertisers

Copyright The U K ATARI COMPUTER OWNERS CLUB is an independent users group and is in no way affiliated with ATARI All rights reserved worldwide. No part of this publication may be reproduced stored in a retrieval system or transmitted in any form or by any means electronic mechanical photocopying recording or otherwise without the prior written permission of the UK ATARI COMPUTER OWNERS CLUB. Whilst reasonable care is taken in the preparation of the magazine the UK ATARI COMPUTER OWNERS CLUB cannot be held legally responsible for any errors that may appear in articles or for any consequences arising therefrom. While every effort is made to ensure the accuracy of articles published no responsibility can be accepted by the UK ATARI COMPUTER OWNERS CLUB or ATARI for the content of any program or article in this publication.

CRACKING THE CODE

Part Thirteen
by Keith Mayhew BSc(hon)

The principles of smooth scrolling, both vertically and horizontally, were described last time. The most important fact is that scrolling can be achieved without the need to move any display data thus making it fast. We continue with some example programs that demonstrate how display lists can be built for scrolling characters such methods form the basis of many games but are also of use in other applications such as word processors.

Vertical Scrolling

Any display list instruction which has its vertical scroll flag bit set will cause its associated line on the display to be shifted upward by the number of scan lines specified by the contents of the VSCROL register.

When a group of consecutive lines have their vertical scroll flag bits set then the whole region will move together as VSCROL is altered. Such a region is terminated by the first line which does not have its vertical scroll flag bit set. The last line actually forms part of the scrolling region. So the smallest possible region must consist of two lines, one with its vertical scroll flag bit set and the next which does not have it set. The last line of any region is displayed as only one scan line in height.

When VSCROL has the value zero then a region is displayed in its normal position with only the top scan line of the last line visible. As VSCROL is increased in value, the region is moved up, losing scan lines from the top of the first line, while bringing in data from the last line. Note that no matter what value VSCROL contains, the region remains exactly the same number of scan lines in height.

Figure 1 illustrates a scrolling region of text and the corresponding display image produced for different values of VSCROL, assuming that each line is normally eight scan lines high.

Once a region has been scrolled upward by its maximum amount using VSCROL, i.e. it is one scan line away from the next line's normal position, then VSCROL should be set back to zero whilst a coarse scroll is performed by adjusting the LMS addresses by adding the number of bytes for one line of data, e.g. 40 bytes for BASIC's GRAPHIC 0 mode. Scrolling the image in the opposite direction is simply a case of decrementing VSCROL and then resetting VSCROL to its maximum value and subtracting from the LMS address. The LMS instruction in the display list will usually be placed on the first scrolling line of a region, determining where the display data will come from.

Listing 1 gives an example of a vertically scrolling region in a display. Listing 2 is the BASIC program to read in the code, after running it, the code can be executed by typing.

The display will show the contents of memory from location 0. By using a joystick the display can be moved up or down over memory while the top line of the display will indicate the address of the first memory location being viewed.

The display list is defined on lines 1900 to 1970 of Listing 1. It consists of a line at the top which displays the text 'Location 0000' and ends with a line which displays the text 'Use joystick to move vertically.' The scrolling lines in the middle of the display list have not been defined but space has been reserved for them to be filled in by the program when it is executed.

After turning off the screen DMA, the program calls on the routine 'CHCONV' to perform the conversion between the ATASCII codes generated by the assembler to the display codes of the character set (as explained last time).

The new display list is then installed and some variables are initialised.

The missing part of the display list is now built with an LMS and address for each line. The addresses are initialised so that each points 32 bytes further on from its predecessor.

The DMA is turned back on with the screen set to the narrow size, which gives 32 bytes per mode 2 line, the VBI routine is enabled and control is returned back to the caller.

During each VBI the joystick is tested to see if it has been moved up or down. If movement has occurred then the scrolling technique described above is used. Note that a copy, or shadow, of the VSCROL register is kept in 'COUNT', this is because the scroll registers are write only.

When a coarse scroll is to be performed then all of the LMS addresses are updated by either adding or subtracting 32. After a coarse scroll the routine 'PRADDR' is called to print the new address of the first LMS at the top of the screen as a four digit hex string. 'PRADDR' uses 'PRHEX' to print each of the two bytes which then calls on 'PRDIGIT' to print each hexadecimal digit to the screen.

Display Boundaries

Just one LMS instruction could have been used for the scrolling region but this would clearly have caused a problem as the display can cross over a 4K boundary. If the display was to be limited to only one 4K block at a time then this approach would be suitable. However, because we want a continuous display, regardless of boundaries, it is necessary to use an LMS instruction on every line.

Choosing the narrow width screen DMA in this example means that each mode 2 line requires 32 bytes. The consequence of this is that multiples of 32 give multiples of 4096, i.e. 32 divides exactly into 4096, so any boundary will always occur at the start of a line. As every line reloads the LMS with the address of the next line, the boundary has effectively been lost!

If a standard width screen was used then, at 40 bytes per line, boundaries would rarely occur at the start of a line and, as we wish to display memory contiguously from line to line, there is no way of avoiding boundary problems. Similarly, a wide screen, at 48 bytes per line, would also cause problems with boundaries.

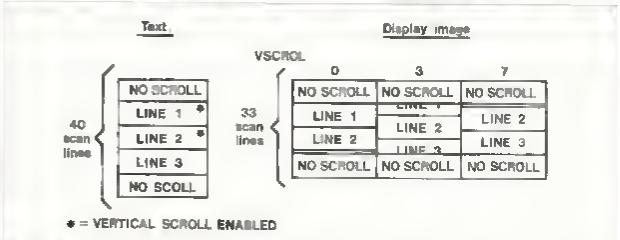


Figure 1.

The big event of 1987...

Atari User Christmas Show

Organised by
Database Exhibitions

**Champagne Suite, Novotel,
Hammersmith, London W6**

10am-6pm Friday November 20
10am-6pm Saturday November 21
10am-4pm Sunday November 22

The record-breaking Atari User Show is back – with three action-packed days devoted to the ever-expanding Atari market.

- ★ **Try out the very latest UK and US 8-bit software**
- ★ **See the full ST range, including the new mega-ST**
- ★ **Snap up the never-to-be-repeated bargains**
- ★ **Expand your system with powerful new add-ons**
- ★ **Get helpful advice from *Atari User's* own experts**
- ★ **Experience the fascinating world of communications**

. . . and much, much more!

This is the one show you mustn't miss!

Send for your advance tickets now – and SAVE £1 a head!

Advance ticket order

Post to: Atari User Christmas Show Tickets, Europa House, 68 Chester Road,Hazel Grove, Stockport SK7 5NY

Please supply

☐ ... tickets at £2 (adults £1) £ ____

☐ ... tickets at £1 (save £1) £ ____

Total £ ____

☐ Cheque enclosed made payable to Database Publications Ltd

☐ Please debit my credit card account

Access / Visa

ATARI USER CHRISTMAS SHOW

Champagne Suite, Novotel, Hammersmith, London W6. November 20-22, 1987

Name

Address

signed

Admission at door £3 (adults), £2 (under 16s)

Advance ticket orders must be received by Wednesday November 11

PHONE ORDERS Ring Show Hotline: 061 480 0171
PRESTEL ORDERS KEY *89, THEN 614568383
MICROLINK ORDERS MAILBOX 72 MAG001 A171
Please quote credit card number and full address + postcode

```
0100 ;Demo of smooth vertical scrolling.
0110 ;Hardware registers...
0120 DMACTL  =      $D400   ;DMA control.
0130 VSCROL  =      $D405   ;Vertical scroll.
0140 ;Operating system shadows...
0150 SDMCTL  =      $022F   ;DMA control.
0160 SDLSTL  =      $0230   ;Display list pointer low.
0170 SDLSTH  =      $0231   ;Display list pointer high.
0180 STICK0  =      $0278   ;Joystick 0.
0190 ;Operating system vectors...
0200 SETVBV  =      $E45C   ;Set VBI vector.
0210 VBI2    =      $E45F   ;VBI stage two vector.
0220 ;Display list equates...
0230 LINES   =      48      ;Number of display lines * 3.
0240 JMP     =      $01     ;Jump.
0250 WSYNC   =      $40     ;Wait for vertical blank flag.
0260 LMS     =      $40     ;Load memory scan flag.
0270 VS      =      $20     ;Vertical scroll flag.
0280 D4      =      $30     ;Blank 4 lines.
0290 D8      =      $70     ;Blank 8 lines.
0300 ;Page zero variables...
0310         *=     $CB
0320 PZERO   *=     *+2     ;Pointer.
0330 COUNT   *=     *+1     ;Scroll counter.
0340         *=     $4000
0350         PLA
0360         LDA    #$00    ;Turn off DMA.
0370         STA    SDMCTL
0380         STA    DMACTL
0390         LDY    #48     ;Correct 48 bytes of text.
0400         LDA    #TITLE&$FF
0410         STA    PZERO
0420         LDA    #TITLE/256
0430         STA    PZERO+1
0440         JSR    CNCONV
0450         LDA    #DLIST&$FF ;New display list.
0460         STA    SDLSTL
0470         LDA    #DLIST/256
0480         STA    SDLSTH
0490         LDX    #0      ;Zero scroll counter.
0500         STX    COUNT
0510         STX    VSCROL
0520         STX    PZERO   ;And LMS pointer.
0530         STX    PZERO+1
0540 LOOP    LDA    #LMS+VS+2 ;Save an instruction.
0550         STA    DATA,X
0560         LDA    PZERO   ;And its address.
0570         STA    DATA+1,X
0580         LDA    PZERO+1
0590         STA    DATA+2,X
0600         LDA    PZERO   ;Move pointer to next line.
0610         CLC
0620         ADC    #32
0630         STA    PZERO
0640         BCC    SKIP
0650         INC    PZERO+1
0660 SKIP    INX            ;Next LMS instruction.
0670         INX
0680         INX
0690         CPX    #LINES-3
0700         BNE    LOOP
0710         LDA    #LMS+2  ;Last one has no scroll flag.
0720         STA    DATA,X
0730         LDA    PZERO
0740         STA    DATA+1,X
0750         LDA    PZERO+1
0760         STA    DATA+2,X
0770         LDA    #$21    ;DMA on with narrow screen.
0780         STA    SDMCTL
0790         LDY    #VBI&$FF ;Install VBI.
0800         LDX    #VBI/256
0810         LDA    #6
0820         JSR    SETVBV
0830         RTS
0840 ;Vertical blank interrupt.
0850 VBI     LDA    STICK0  ;Test for up or down movement.
0860         AND    #1
0870         BEQ    UP
0880         LDA    STICK0
0890         AND    #2
0900         BEQ    DOWN
0910         JMP    VBEXIT
0920 UP      DEC    COUNT   ;Up - decrement scroll count.
0930         BPL    UFINE   ;If positive then fine scroll.
0940         LDA    #7      ;Reset count.
0950         STA    COUNT
0960         LDX    #0      ;Adjust all LMS bytes.
0970 UPLOOP  LDA    DATA+1,X
0980         SEC
0990         SBC    #32
1000         STA    DATA+1,X
1010         BCS    USKIP
1020         DEC    DATA+2,X
1030 USKIP   INX
1040         INX
1050         INX
1060         CPX    #LINES
1070         BNE    UPLOOP
1080         JSR    PRADDR  ;Print new LMS address.
1090 UFINE   LDA    COUNT   ;Update fine scroll value.
1100         STA    VSCROL
1110         JMP    VBEXIT  ;Exit.
1120 DOWN    INC    COUNT   ;Down - increment fine count.
1130         LDA    COUNT
1140         CMP    #8
1150         BNE    DFINE   ;If not max. then fine scroll.
1160         LDX    #0      ;Reset fine scroll count.
1170         STX    COUNT
1180 DNLOOP  LDA    DATA+1,X ;Adjust all LMS bytes.
1190         CLC
1200         ADC    #32
1210         STA    DATA+1,X
1220         BCC    NSKIP
1230         INC    DATA+2,X
1240 NSKIP   INX
1250         INX
1260         INX
1270         CPX    #LINES
1280         BNE    DNLOOP
1290         JSR    PRADDR  ;Print new LMS address.
1300 DFINE   LDA    COUNT   ;Update fine scroll.
1310         STA    VSCROL
1320 VBEXIT  JMP    VBI2    ;Back to O.S.
1330 ;Print out address of first LMS.
```

```
1340 PRADDR  LDA     #NUMBER&$FF ;Copy address of number.
1350         STA     PZERO
1360         LDA     #NUMBER/256
1370         STA     PZERO+1
1380         LDY     #0          ;Zero index for high byte.
1390         LDA     DATA+2      ;Get high byte.
1400         JSR     PRHEX       ;Print it.
1410         LDY     #2          ;Index to low pair digits.
1420         LDA     DATA+1      ;Get low byte.
1430         JSR     PRHEX       ;Print it.
1440         RTS                 ;Return.
1450 ;Print a byte in hex format.
1460 PRHEX   PHA                 ;Save A.
1470         LSR     A           ;Get high nibble.
1480         LSR     A
1490         LSR     A
1500         LSR     A
1510         JSR     PRDIGIT ;Print it.
1520         PLA                 ;Restore A.
1530         AND     #$0F        ;Mask for low nibble.
1540         INY                 ;Next character index.
1550         JSR     PRDIGIT ;Print it.
1560         RTS                 ;Return.
1570 ;Print a single hex digit.
1580 PRDIGIT CMP     #$0A
1590         BCC     DIGIT       ;Below 10 is digit.
1600         CLC                 ;Else add to get char. code.
1610         ADC     #$17
1620         JMP     SAVCHR
1630 DIGIT   CLC                 ;Digit - Adjust it.
1640         ADC     #$10
1650 SAVCHR  STA     (PZERO),Y ;Save character code.
1660         RTS                 ;Return.
1670 ;Convert an ATASCII string to display characters.
1680 CHCONV  DEY                 ;Y holds number of characters.
1690         LDA     (PZERO),Y ;Get character.
1700         TAX                 ;Save it.
1710         AND     #$7F        ;Turn off high bit (inverse).
1720         CMP     #$60
1730         BCC     CH1
1740         TXA                 ;Above $60 - no adjustment.
1750         JMP     CHOK
1760 CH1     CMP     #$20
1770         BCC     CH2
1780         TXA                 ;Between $20 and $5F.
1790         SEC                 ;Subtract $20.
1800         SBC     #$20
1810         JMP     CHOK
1820 CH2     TXA                 ;Between $00 and $1F.
1830         CLC                 ;Add $40.
1840         ADC     #$40
1850 CHOK    STA     (PZERO),Y ;Save character back.
1860         CPY     #$00        ;Last character?
1870         BNE     CHCONV
1880         RTS                 ;Return.
1890 ;Display list data.
1900 DLIST   .BYTE   $0,$6,$8,LMS+7
1910         .WORD   TITLE       ;Title string address.
1920         .BYTE   $4
1930 DATA    *=      *+LINES ;Filled in by program...
1940         .BYTE   $4,LMS+2
1950         .WORD   FOOTER  ;Footer string address.
1960         .BYTE   JMP+WVB
1970         .WORD   DLIST
1980 TITLE   .BYTE   ' LOCATION: $0000 '
1990 NUMBER  =       TITLE+11 ;Address of first '0' in title.
2000 FOOTER  .BYTE   'Use joystick to move vertically.'
```

Listing 1

```
RZ 10 DIM HEX$(16)
HX 20 LINE=10000:TRAP 100:J=0:START=24576
VA 30 READ HEX$,CHKSUM:SUM=0
AA 40 FOR I=1 TO 15 STEP 2
ZB 50 D1=ASC(HEX$(I,I))-48:D2=ASC(HEX$(I+
   1,I+1))-48
KT 60 NUM=((D1-7*(D1)16))*16+(D2-7*(D2)16
   ))
LN 70 SUM=SUM+NUM:POKE START+J,NUM:J=J+1:
   NEXT I
LY 80 IF SUM=CHKSUM THEN LINE=LINE+10:GOT
   O 30
IN 90 ? "Checksum error on this line:"
VQ 95 LIST LINE:END
YB 100 PRINT "Data in memory."
RA 10000 DATA 68A900802F020D00,604
KQ 10010 DATA D4A030A96C85CBA9,1202
VH 10020 DATA 6185CC200A61A92E,769
JT 10030 DATA 8D3002A9618D3102,649
ZY 10040 DATA A2008AC09E85D486,994
MF 10050 DATA C86ACCA9629D3561,1115
YI 10060 DATA A5CB9D3661A5CC9D,1202
QD 10070 DATA 3761A5CB18692085,814
AC 10080 DATA C09002E6CCE0E8E8,1479
MA 10090 DATA E02DD0DFA9429D35,1145
NG 10100 DATA 61A5CB9D3661A5CC,1142
US 10110 DATA 9D3761A9218D2F02,701
UW 10120 DATA A06AA260A9062B5C,823
LU 10130 DATA E460AD70022901F0,901
YX 10140 DATA 0AA070022902F020,633
SB 10150 DATA 4CCE60C6CD1D1EA9,996
KS 10160 DATA 0785CDA200E03661,847
NB 10170 DATA 38E92009D366180B3,800
EU 10180 DATA DE3761E0E0E0E030,1342
YS 10190 DATA D0E020D160A5CD8D,1291
KR 10200 DATA 05D44CCE60E6CDA5,1195
MD 10210 DATA CDC908D01CA200B6,946
NC 10220 DATA CD0D3661186920 9D,863
SH 10230 DATA 3661900 3FE3761E6,936
XF 10240 DATA E0E0E030D0EB20D1,1420
AA 10250 DATA 60A5CDB0D5D44C5F,995
YT 10260 DATA E4A97705C0A96105,1251
VX 10270 DATA CCA000AD376120EA,955
IJ 10280 DATA 60A002AD366120EA,840
MB 10290 DATA 6060404A4A4A4A20,592
DY 10300 DATA FA606E290FC020FA,900
MP 10310 DATA 6060C90A90061069,682
TL 10320 DATA 174C07611869091,493
BR 10330 DATA C660E61C6AA297F,1153
WJ 10340 DATA C9609004B4C2761,795
TJ 10350 DATA C920900706A3EE920,843
BC 10360 DATA 4C27610A18694091,668
CZ 10370 DATA C8C000D0D0607070,1144
RZ 10380 DATA 70476C6130000000,436
HS 10390 DATA 0000000000000000,0
ML 10400 DATA 0000000000000000,0
MP 10410 DATA 0000000000000000,0
MT 10420 DATA 0000000000000000,0
MX 10430 DATA 0000000000000000,0
TU 10440 DATA 00000000030427C,230
TL 10450 DATA 61412E612D4C4F43,559
KB 10460 DATA 4154494F4E3A2030,517
VM 10470 DATA 3030302055736520,589
JG 10480 DATA 6A6F797374696368,800
YX 10490 DATA 20746F206D6F7665,730
VD 10500 DATA 2076657274696361,782
TG 10510 DATA 6C6C792E,393
```

Listing 2.

Does this mean that to scroll correctly over a 4K byte boundary that a narrow screen must be used? The answer is no, the reason why the wider screens could not be used in our example is that contiguous memory is being displayed from line to line, however, if we use noncontiguous memory for the display then this problem is easily overcome.

There are two basic approaches which can be used to display a continuous image whilst crossing boundaries.

The most obvious method is illustrated in Figure 2 this has continuous image data starting at the first boundary and extending to just before the next boundary Once the screen has been scrolled into the last part of this image then the display is switched to the start of the next boundary By having a copy of the previous screen's image at the start of the next block means that the display will appear continuous as the next block is brought into view The advantage of this method is that only one LMS instruction is needed but approximately one screen's image data has to be duplicated for each new block.

The second method requires LMS instructions on every line but each picks up data spaced at, say, 64 bytes apart so that any boundary will always coincide with the start of a line Figure 3 illustrates how the display image comes from tha

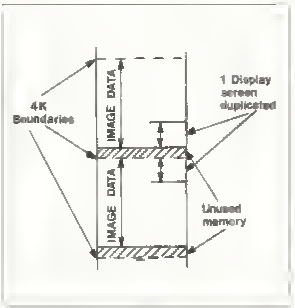


Figure 2

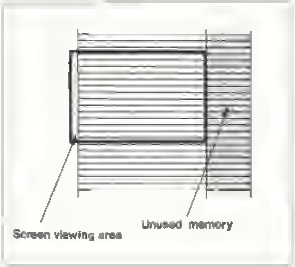


Figure 3.

first part of each line with some unused memory at the end of each line Unlike the previous method, the data has to be noncontiguous because of the unused memory on each line

Horizontal Scrolling

As with vertical scrolling, having the horizontal scroll flag bit set for a line, causes it to be moved to the right by the number of colour clocks specified by the contents of the HSCROL register Note, that as scrolling is done by colour clocks, the HSCROL register only extends from 0 to 3 for a standard character cell, such as mode 2, not 7!

Once a line has been shifted by its maximum amount then HSCROL should be reset to zero and one added to the address of the LMS instruction for that line Reversing this process produces scrolling to the left instead of the right To scroll a region of lines horizontally requires each to have its horizontal scroll flags set and each to have its own LMS instruction

Figure 3, which shows the set-up for a vertically scrolling screen, also serves as an illustration of horizontal scrolling over several lines. the line length might be anything up to 4K bytes As with vertical scrolling, it is necessary to arrange data so that boundaries occur at the start of a line

```
0100 ;Demo of smooth horizontal scrolling.
0110 ;Hardware registers...
0120 DMACTL  =      $D400   ;DMA control.
0130 HSCROL  =      $D404   ;Horizontal scroll.
0140 ;Operating system shadows...
0150 SDMCTL  =      $022F   ;DMA control.
0160 SDLSTL  =      $0230   ;Display list pointer low.
0170 SDLSTH  =      $0231   ;Display list pointer high.
0180 ;Operating system vectors...
0190 SETVBV  =      $E45C   ;Set VBI vector.
0200 VBI2    =      $E45F   ;VBI stage two vector.
0210 ;Display list equates...
0220 JMP     =      $01     ;Jump.
0230 WVB     =      $40     ;Wait for vertical blank flag.
0240 LMS     =      $40     ;Load memory scan flag.
0250 HB      =      $10     ;Horizontal scroll flag.
0260 B4      =      $30     ;Blank 4 lines.
0270 B8      =      $70     ;Blank 8 lines.
0280 ;Page zero variables...
0290         *=     $CB
0300 PZERO   *=     *+2     ;Pointer.
0310 PDS     *=     *+1     ;Position in string.
0320 COUNT   *=     *+1     ;Scroll counter.
0330         *=     $6000
0340         PLA
0350         LDA    #$00    ;Turn off DMA.
0360         STA    SDMCTL
0370         STA    DMACTL
0380         LDY    #20     ;Correct strings.
0390         LDA    #STAAS&#FF
0400         STA    PZERO
0410         LDA    #STAAS/256
0420         STA    PZERO+1
0430         JSR    CHCONV
0440         LDY    #MESSLEA
0450         LDA    #MESSAGE&#FF
0460         STA    PZERO
0470         LDA    #MESSAGE/256
0480         STA    PZERO+1
0490         JSR    CHCONV
0500         LDY    #43     ;Copy start of message to end.
0510 CPYMESS LDA    MESSAGE,Y
0520         STA    COPY,Y
0530         DEY
0540         BPL    CPYMESS
0550         LDA    #DL1ST&#FF ;New display list.
0560         STA    SDLSTL
0570         LDA    #DL1ST/256
0580         STA    SDLSTH
0590         LDA    #0
0600         STA    COUNT   ;Zero fine scroll counter.
0610         STA    PDS     ;Zero position in string.
0620         STA    HSCADL  ;Reset hardware scroll.
0630         LDA    #$22    ;DMA on with standard screen.
0640         STA    SDMCTL
0650         LDY    #VBI&#FF ;Install VBI.
0660         LDX    #VBI/256
0670         LDA    #6
0680         JSR    SETVBV
0690         RTS
```

```
0700 ;Vertical blank interrupt.
0710 VBI    DEC   COUNT   ;Decreasnt scroll count.
0720        BPL   FINE    ;If posltive just fine scroll.
0730        LDA   #3      ;Else reset to eax. value.
0740        STA   COUNT
0750        INC   POS     ;Increasnt posltion.
0760        LDA   POS
0770        CMP   #MESSLEN ;At end?
0780        BNE   NEXT    ;No, go to next character.
0790        LDA   #MESSAGE&$FF ;Yes, go to start.
0800        STA   DATA
0810        LDA   #MESSAGE/256
0820        STA   DATA+1
0830        LDA   #0
0840        STA   POS
0850        JMP   FINE    ;Update fine scroll.
0860 NEXT   INC   DATA    ;Go to next character.
0870        BNE   FINE
0880        INC   DATA+1
0890 FINE   LDA   COUNT   ;Update fine scroll.
0900        STA   HSCROL
0910 VBEXIT JMP   VBI2    ;Back to O.S.
0920 ;Convert an ATASCII string to display characters.
0930 CHCONV TAY           ;Y holds number of characters.
0940        LDA   (PZERO),Y ;Get character.
0950        TAX           ;Save it.
0960        AND   #$7F    ;Turn off high bit (inverse).
0970        CMP   #$60
0980        BCC   CH1
0990        TXA           ;Above $60 - no adjustment.
1000        JMP   CHOK
1010 CH1    CMP   #$20
1020        BCC   CH2
1030        TXA           ;Between $20 and $5F.
1040        SEC           ;Subtract $20.
1050        SBC   #$20
1060        JMP   CHOK
1070 CH2    TXA           ;Between $00 and $1F.
1080        CLC           ;Add $40.
1090        ADC   #$40
1100 CHOK   STA   (PZERO),Y ;save character back.
1110        CPY   #$00    ;Last character?
1120        BNE   CHCONV
1130        RTS           ;Return.
1140 ;Display list data.
1150 DLIST  .BYTE 8*8,8*8,LMS+7
1160        .WORD STARS   ;Top string.
1170        .BYTE 4*LMS+HS+2
1180 DATA   .WORD MESSAGE ;Scrolling message.
1190        .BYTE 4*LMS+7
1200        .WORD STARS   ;Bottom string.
1210        .BYTE JMP+JVB
1220        .WORD DLIST
1230 STARS  .BYTE '**********************'
1240 MESSAGE .BYTE '    This is a demo of smooth '
1250        .BYTE 'horizontal scrolling. This message '
1260        .BYTE 'goes on and on and on...'
1270 MESSLEN =    *-MESSAGE ;Number of bytes in string.
1280 COPY   *=    *+44    ;Space for copy of string.
```

Listing 3.

Listing 3 is a program which sets up a display list to scroll a single line of mode 2 characters in a continuous loop like a message board. After adjusting the text string for display, the first 44 characters are copied to just after the end of the message string. this corresponds to the data duplication for vertical scrolling as in Figure 2.

Once the VBI has been installed the message is fine scrolled to the left. The coarse scrolls are counted by the variable 'POS'. When 'POS' reaches its maximum value it means that the whole message has been displayed and the copy of the start of the message is now in view. The LMS address is reset at this stage to point back to the start of the string which is identical to the image already being displayed - so the loop can continue, making the message look infinitely long. Listing 4 is the BASIC program to read in the code for this program, it is executed by typing

X=USR(24576)

Screen Widths

The bottom two bits of DMACTL determine the width of the screen, i.e. how many bytes will be fetched for a line of data. The narrow screen (bit 1 set to 0 & bit 0 set to 1) has a total of 128 colour clocks in width. The standard screen (bit 1 set to 1 & bit 0 set to 0) has 32 more colour clocks, giving it a total of 160. Lastly, the wide screen (bit 1 set to 1 & bit 0 set to 1) has a further 32 colour clocks, giving it a total of 192.

This information is useful in the context of horizontal scrolling because ANTIC actually fetches different amounts of data for a line when its horizontal scroll flag bit is set. For a narrow screen the amount of data fetched for a scrolling line is the same as that for a standard screen. A standard screen uses the same amount as a wide screen. A wide screen, however, does not cause any extra data to be fetched for a scrolling line.

For the narrow and standard screen widths, scrolling lines take their data from 16 colour clocks further in than usual. Thus you must arrange your data accordingly. As HSCROL is increased in value then this hidden data at the left edge of the screen starts to become visible.

In wide screen mode the situation is much simpler. There are no offsets for data to be accounted for as data is displayed as normal. As HSCROL is increased in value then background colour is scrolled in to fill up the gap at the left edge. This is really of little consequence, though, as the extremes of the picture to the left and right are usually well out of visibility on most monitors and televisions.

What Next?

Having seen how horizontal and vertical scrolling can be used all that remains is for you to make use of these techniques in your programming.

It should be obvious that horizontal and vertical scrolling can be mixed at will and it is very easy to produce a screen which scrolls in all directions over a background, or landscape, which is used as the basis to many games. Of course multicolour character sets make a world of difference over boring old mode 2!!! The use of DLI's to produce areas scrolling at different speeds or directions is also quite effective. Just add some players and missiles and you will have exploited much of the power of these fine machines. Above all, experiment, it is the only real way to learn!

Next Time

Yes, this series will undoubtedly be back next time!

Over thirteen parts we have covered just about everything from binary

arithmetic to a detailed look at the hardware features for multicolour graphics. If you have joined this series rather late and have missed much of the valuable information then do not hesitate to send for back issues and don't forget to get copies for your friends either!

In the future we still have some more mundane aspects of the hardware to cover as well as the use of the sound channels. However the main area of discussion is going to be the operating system. I have little idea what will be presented next time at this stage so you will just have to wait to be surprised!

Editors note. In part twelve of this series the word 'luminescence' was inadvertently substituted for 'luminance', a small but significant error, sorry for any misunderstanding it may have caused.

```
OZ 10 DIM HEX$(16)
HX 20 LINE=10000:TAAP 100:J=0:STAAT=24576
VA 30 HEAD HEX$,CHKSUM:SUM=0
AA 40 FOR I=1 TO 15 STEP 2
ZG 50 D1=ABC(HEI$(I,I))-48:D2=ABC(HEI$(I+
   1,I+1))-49
KT 60 NUM=(10I-7*(D1)16))*16+(D2-7*(D2)16
   ))):
LW 70 SUM=SUM+NUM:POKE GTAAT+J,NUM:J=J+1:
   NEXT I
LY 80 IF SUM=CHKSUM THEM LINE=LIWE+10:GOT
   O 30
IN 90 ? "Chockeue error on thin line:"
VO 95 LIST LIWE:END
YS 100 PAINT "Data in eeaory."
AA 10000 DATA 6AA900ED2F028D00,604
WI 10010 DATA D4AB14A98685C8A9,1248
QL 10020 DATA 6085CC20B160A050,939
FA 10030 AATA A9CA85C8A960B5CC,1309
JW 10040 AATA 20B160A02BB9CA60,943
IC 10050 DATA 9922619810F7A9A5,1017
IF 10060 DATA 8D3002A960D3102,640
LB 10070 DATA A90085CE85CDED04,991
BJ 10080 DATA D4A9220D2F02A050,845
VY 10090 DATA A260A9062B5CE460,861
BZ 10100 DATA C6CE1025A90385CE,968
BL 10110 DATA E6CDA5CDC955D011,1319
EO 10120 DATA A9CA8DAD60A960ED,1187
IG 10130 DATA AE60A900B5CD4C79,974
JI 10140 DATA 60EEAD60D0D3EEAE,1226
ZT 10150 DATA 60A5CE60D04D44C3F,995
QL 10160 DATA E4980ICBAA297FC9,1203
NB 10170 DATA 6A990A6A4C9E60C9,913
YV 10180 DATA 2B90B70A3BE9204C,718
LW 10190 DATA 9E60BA18694091C0,933
IF 10200 DATA C0B0D0D6B7B7B7B0,1053
DO 10210 DATA 47B6603452CA6030,825
BE 10220 DATA 47066041A5602A2A,759
VO 10230 DATA 2A2A2A2A2A2A2A2A,336
VS 10240 DATA 2A2A2A2A2A2A2A2A,336
WI 10250 DATA 2A2A2B2B2B2B546E,480
AF 10260 DATA 6973206973206120,633
FU 10270 DATA 64656D6F206F6620,698
YC 10280 DATA 7363GF6F74602069,802
AL 10290 DATA 6F72697A6F6E7461,886
WD 10300 DATA 6C207363726F6C6C,795
BB 10310 DATA 696E672E20546869,689
JE 10320 DATA 7320636573736167,787
PD 10330 DATA 6520676F6573206F,706
DZ 10340 DATA 6E20616E64206F6E,702
QG 10350 DATA 20616E64206F6E2E,638
BP 10360 DATA 2E2E,92
```

Listing 4

2 BIT SYSTEMS: MASTERS OF SOUND

PRODUCTS FOR THE 48K ATARI

REPLAY

Replay is a complete sound sampling system that allows you to produce real speech music on any 8-Bit ATARI

Features:

Sample rate selectable from 6Khz to 21Khz
Sample playback through TV/Monitor
Allows samples to be used in your own Basic programs
Supplied on cartridge, no user memory lost
Records from Hi-Fi or external recorder
V2.0 Replay program with reverse and trigger options
Also included in the REPLAY PACKAGE

Digidrum: Digital Drum sequencer (no hardware required) allows you to create your own Drum rhythms using 8 sampled drum sounds

Digisynth: Simple sample sequencer (no hardware required) allows you to play tunes using sampled sounds (Pop beats, guitars, voices etc.)

Echo: A real time Echo program to create space, beacie Delay variable between 2mS to 1 sec

PRICE:
REPLAY system (cartridge, software, Digidrum and Digisynth) only £39.95

NEW MICRO-VOICE

Micro-Voice is a low cost sampling package that allows you to record play sounds on any 8 bit Atari

Features:

Samples can be played back through TV/Monitor
Can plugs into joystick port
Records from a Tape recorder/HI-FI
Allows samples to be played in your own BASIC programs

PROGRAMS INCLUDED IN THE MICRO-VOICE PACKAGE

MICRO-SEQ: A step time sequencer

MICRO-SYNTH: Allows you to play tunes using samples/synths

ECHO: A real time echo program

SAMPLER: Allows you to record/Edit samples

USER: Basic routines

PRICE:
MICRO-VOICE system (cable and above software) only £19.95

MIDI MASTER

MIDI Master is a MIDI interface for your ATARI allows you to take full control of synths etc

Features:

MIDI In/out cables for easy connection
8 Track Real time sequencer with tempo correction
Cassette CZ set up voice editor (allows you to edit and store voices on Disk/Tape)
Yamaha DX100/21 series voice editor
Music player program (allows you to play Music composed on AMS2 files via MIDI)
New voices used DX7 voice editor
CZ menu and split program
NB We will be producing other voice editors for different synths as if you own a different synth get in touch

FUTURE RELEASES:
To enhance MIDI MASTER to include a 16 track sequencer

PRICE:
MIDI MASTER interface CABLE plus software) only £29.95

DISK PRICE CRASH

10 DISK BOX, INCLUDING LABELL AND TAB ONLY £4.95

PERCUSSION MASTER

A high quality professional drum kit for the ATARI

Features:

External D/A and filter to ensure high quality sound which plugs into joystick ports 1 & 2
8 Sampled Drum sounds
Polyphonic Rhythm editor using pull down menus and windows
Capability to load new drum sounds from Disk
* Includes enhanced version of REPLAY software allowing REPLAY owners to sample sounds with a far greater resolution
Includes digital echo program for use with the REPLAY cartridge
3 Channel polyphonic
100 Song entries
Audio output via hi-fi

PRICE:
PERCUSSION MASTER hardware & software £29.95

PERCUSSION MASTER SAMPLE DISK 1
Contains over 30 separate samples
PRICE ONLY £5.95

DIGIDRUM II

A digital drum machine/sample sequencer that requires no extra hardware

Features:

8 sampled drum sounds
Step time pattern entry
Capability to load Replay samples into any of the 8 sample positions
KITS can be stored or saved to disk or cassette
2 Channel polyphonic (any two Drums/1 Song entries with comp to keep right)
32 Patterns (4-32 Beats/Bar)

PRICES:
DISK £6.95
TAPE £5.95

DIGIDRUM II/REPLAY SAMPLE DISK 2

A brutally added 8 bit containing over 50 (8KHz samples that can be loaded into the program and saved as a KIT for future use
PRICE ONLY £4.95

ORDER FORM

Name

Please send me

Address

- ☐ Replay System on Disk/Tape*
- ☐ Dig drum II on Disk/Tape
- ☐ Midi Master Interface on Disk/Tape
- ☐ Percuss on Master System Disk ONLY
- ☐ Micro-Voice on disk
- ☐ Digidrum II/Replay Sample Disk 1
- ☐ Percussion Master Sample Disk 1
- ☐ 10 DS/DD Disks

Postcode

I enclose a Cheque/Postal Order for £ (Payable to 2 Bit Systems)

* Delete as applicable

Send to: **2 Bit Systems Ltd., 395, Hitchin Rd., Luton, BED6**

EIGHT BIT SOFTWARE

Software Librarian - Roy Smith

There are two ways to get programs from the library. You can use the donation scheme by sending in a disk or cassette of your own, or if you have a program of your own which you would like to add to the library you can exchange it for 3 programs of your choice. The rules are as follows

3 FOR 1 EXCNANGE

1 Every program you donate entitles you to three programs in return

2 The program you donate must be your own original and not copied

3 Your donated program must be submitted on a cassette or a disk, programs in the form of print outs cannot be processed

4 If your program requires any special instructions they should be added in the form of REM statements within the program (or you may present them as instructions when the program is actually run)

5 BONUS Every program submitted per quarter (between issues of the magazine) will be eligible to be judged 'STAR PROGRAM' for that quarter. This carries a prize of £10 which will be paid to the author. The programs will be judged by the Editorial Team and their decision is final. The Editorial Team are not eligible for the prize.

6 Please include 30p in stamps (or cash) to cover return postage

7 The '3 for 1' exchange is only open to club members

DONATION SCHEME

1 Every club member can make a donation to the club, at any time, if he/she wishes to obtain a particular program(s)

2 There is no limit on the number of programs that can be asked for at any one time. (If you are asking for a lot of programs at once, please ensure that you send a sufficient number of disks or cassettes. It's better to send too many than not enough.)

3 Please include 30p in stamps (or cash) minimum to cover return postage. If your parcel costs more than 30p to send to us, please include an amount equal to that of the postage, so that we may return your parcel to you without delay. Overseas members should add an extra £1 to cover postage costs

4 The donation fee is £1 per program. Cheques or Postal Orders should be made out to the U.K. Atari Computer Owners Club

5 You should send in blank disks or cassettes, ensuring they are properly packed to prevent damage in the post. State which programs you require and remember to give your name and address. Also remember to include the fee and return postage.

6 The 'Donation Scheme' is only open to club members.

The Library Software Service is for subscribers only

LIBRARY SOFTWARE TITLES

Listed below are the software titles received by members for inclusion in the library since the last issue was published. As the library now contains over 350 programs, it is getting too big to print the entire list. For those of you who are new to Monitor and are unaware of what is available, then send for a photocopy of the complete list which is available from the librarian. There is a small charge for this service to cover photocopying costs. If you would like a list send 50p and a S.A.E. for return.

Games

BERG

by Steve Hillen Southend
Excellent adventure set in the icy waters of the North Atlantic.
Runs in 32K min Disk only

CONNECT TO FOUR

by S Fawcett Canterbury
Good example of the well known board game
Runs in 16K min Disk or Cassette
400/800 only

Nome Entertainment

SCRABBLE CROSSWORD

by Terry Wood Reading
A combination of a crossword and a scrabble type game. Very entertaining
Runs in 48K min Disk or Cassette

DOMINO KID

by Terry Wood Reading.
Have a game of dominos with the children
Runs in 32K min Disk or Cassette

Demos

GAD PICTURE SHOW

by Lawrence Hoare Reading
A selection of pictures created using Graphic Art Designer
Runs in 48K min Disk only
Requires 1 side of a disk.

Utilities

AUTOBOOT CAS MAKER

by Adrian Cox Reading.
Make autoboot cassettes of binary files or Assembler/Editor cassette object files
Runs in any size. Disk only

MO2 CONVERTER

by Adrian Cox Reading.
Strips the header from a Mini Office II word processor file.
Runs in any size Disk only

MASTER DISK

by Mark Furnell Telford.
A selection of useful back up programs
Runs in any size Disk only
XL/XE systems only

MORSE CODE

by Mark Furnell Telford
Simulates sending and receiving morse code
Runs in any size. Disk or Cassette.

EDUCATING SAM

by S Fawcett Canterbury
A letter and number recognition program for use with SAM
Runs in 16K Disk or Cassette

RAMDISK MOVER

by Trevor Peart Jackson High Wycombe.
On booting up your 130XE a disk directory is displayed and a file can be moved into RAMdisk.
Runs in any size. Disk only
130XE only

*** STAR PROGRAM ***

SECTOR EDITOR

by Gary Cheung Harold Wood
Excellent disk sector editor written in Action! (not required)
Runs in any size Disk only

****

TURBO COMPILER

If you have Turbo Basic you'll definitely want the compiler!
Runs in any size Disk only.
XL/XE only
One side of a disk required.

Education

ALPHABET

by Terry Wood Reading.
An alphabet learning aid with scrolling pictures and words
Runs in any size Disk only

BERG BERG BERG

by STEVEN HILLEN Runs in 32K

This adventure column is a little different to previous ones in that I'm not going to explain how the program listed works It is a complete adventure that you can type in and play

It uses the data structures detailed in issue 12 the parser from issue 14 and screen display ideas from issue 16 I did not include the text compression routines from issue 13 as this makes the typing pretty tedious Memory space was not a major consideration here anyway as the program fits inside any 48K or 64K machine

Type in the listing, and save it off to cassette or disk The program can be loaded back as for standard BASIC files The program recognises words by their first 5 letters and can understand sentences such as 'HIT THE NAILS WITH THE HAMMER' although in most cases the simple verb/noun type structure is sufficient, e g 'TAKE BOOK Since the game has a nautical flavour, directions are (F)orward, (A)ft, (P)ort, (S)tarboard, (U)p and (D)own

OK then the scene is set You are on a luxury transatlantic liner, quite near the Arctic As usual, you are taking a late night stroll around the decks prior to going to bed As usual you have been at the bar all night and right now are feeling a little squiffy Suddenly the liner turns, and you stagger to the guardrail, and clumsily topple overboard Nobody noticed and the liner steams off into the distance Luckily, the icy water has sobered you up You'll need to be able to think straight to get out of this one!

NOTE. In this program, anything which is underlined should be entered in 'INVERSE'

```
JB 1 GRAPHICS 0:? :?
KO 10 GOTO 9000
OO 100 X=USR(ML2,N+C,L-C,32):Y=USR(ML2,R+
   C+X,L+5-C-X,32):IF C+X+Y)L THEN POP :G
   OTO 220
YO 110 K$(C1,5)="     ":K$=IN$(X+C,Y+C+X-
   1):C=C+Y+X:IF USR(ML1,RDN(U$),C1,NDR(K
   $)) THEN 100
ZO 120 RETURN
MA 130 REM ********************************
GJ 140 TRAP 230:C=C1:? "What now -->";:IN
   PUT 1N$:R=NDN(1N$)-C1:L=LEN(1N$)+C1:IN
   $(L,L)=" "
NO 145 V=C0:N=C0:P=C0:1NN=C0
NS 150 GOSUB GN:V=USR(ML1,NDR(V$),NV,ROR(
   K$)):IF  NOT V THEN ? "Is ";K$;" a vsr
   b ?":GOTO 230
UG 160 GOSUB GN
1U 170 N=USR(ML1,NDB(N$),NNN,ROR(K$)):IF
   N THEN N=N:GOTO 160
FT 180 N=USR(ML1,RDB(P$),NP,NDN(K$)):IF
   NOT N THEN ? "What doss ";K$;" esan ?"
   :GOTO 230
SU 190 P=N:GOSUB GN:N=USR(ML1,ADN(N$),NNN
   ,NDN(K$)):IF N THEN IMN=N:GOTO 160
F0 200 REM **
YY 220 GOSUB 1000+100+V
OW 221 IF CL(4 THEN RIR=RIR-C1:IF  NOT RI
   N THEN ? "You have died of the cold.":
   GOTO LOSE
SC 225 GOSUB 300
PH 230 GOSUB NAME:GOTO 140
FY 300 REM +JNCK+
UN 310 GOTO 300+JACK*50
ZJ 350 RETURN
6H 400 IF RND(0)>0.09 OR CL=32 THEN RETUR
   N
QN 405 ? "N straggly bearded old pirate w
   anders through and with unbslievable s
   pssd  swipes all your possssslons."
CL 404 Z=C0:? "He's vanished !";XX=C0:IF
    NOT RSC(O$(34,34)) THEN Z=C1
HZ 407 C=USR(ROR(ML4$),ADR(O$)-C1+XX,39-X
   X,C0)
GL 400 IF C THEN O$(C+XX,C+XX)=CHR$(26):N
   OBJ=NOBJ-C1:IF X<NNN THEN 407
QF 409 IF Z THEN ? "Luckily, hs esssed yo
   ur key.":O$(34,34)=CHN$(CD):NOBJ=NOBJ+
   C1
TH 410 IF ST$(C1,C1)="R" THEN ? "But hs's
    dropped sosething !":DOUB=2:O$(32,32)
   =CHR$(CL):ST$(C1,C1)="0":RETURN
TE 412 ? :? "You hear hie say : ' Rha, sy
    doubloon is here."
ZU 414 IF ST$(C1,C1)="0" THEN ? "Sot :t b
   ack. No bo ho.":OT$(C1,C1)="R":O$(32,3
   2)=CHN$(99):RETURNN
GG 416 IF ST$(C1,C1)="C" THEN JNCK=2.5:?
   "'It's in this box.' Jack climbs into
   ths box.":O$(32,32)=CHR$(99):RETURN
KK 425 ? "'Found es doubloon eateys.' ":DO
   UB=DOUB-C1
VP 430 IF  NOT DOUB THEN JRCK=2:ST$(C1,C1
   )="R":O$(32,32)=CHR$(99):? "Jack grabs
    his doubloon and vanishss  again."
ZI 440 NETURNN
GG 450 IF RND(C0)<0.3 THEN ? "Bang bang b
   ang. Jack is furious, but  the box hol
   ds."
ZZ 455 RETURN
NC 500 ? "Jack appsars in ths rooe - hs's
    sad  with angsr. He rushes towards y
   ou,   brandishing a sword."
EV 502 D$(C1,C1)=CHN$(99)
LC 505 ? "His face is a hideous eask. He'
   ll killyou. Jack stabs you through the
    chest.":JRCK=5:D$(8,8)=CHR$(CL)
ON 506 IF ST$(2,2)="N" THEN ? "Your life
   fades away.":GOTO LOSE
C0 510 ? "It doss not penetrate. Jack sta
   ggers  back, eyes widened with shock."
   :NETURNN
TO 550 ? "Jack sakss no eistake this tise
   . He  backs s wild stroke to your hea
   d...  Goodbye.":GOTO LOSE
ZC 600 RETURN
U0 700 ? :? :? "You died with a score of
   "|SCONE|"/100"
JR 710 ? :? "Type any key to start again.
   ..":OPEN #C1,4,C0,"K:":GET #C1,K:CLOS
   E #C1:RUN
ZN 720 ? "Well dons. You've finished the
        adventure with a score of ";SCO
   RE;"/100":GOTO 710
XI 750 SCONE=SCORE+10:RETURN
TH 890 IF  NOT N THEN POP :? "What is ths
   s ohject ?":RETURN
RL 895 NETURNN
K0 900 IF  NOT NL THEN ? "You can't move
   that way.":? :RETURN
00 905 R$(CL*6-5,CL*6)=CL$(C1,6)
NQ 910 CL=NL:CL$(CL*6-5,CL*6)=N$(CL*6-5,CL*6)
ZL 920 XX=C0:S=CL:GOSUB DESC:? "You can s
   se the following :":GOSUB O0J:RETURN
0C 930 RESTORE B200+24*CL:READ T$:? T$:RER
```

```
   0 T$:? T$:RETURN
KB 940 RESTORE 8000+CL:READ T$:CY=PEEK(84
   ):CX=PEEK(85)
BE 945 POKE 752,1:IZ=USR(ADR(RL$))
WP 950 POSITION 5,CB:? T$:POSITION 31,CB:
   ? 8CONE;"/100":POKE 84,CY:POKE 85,CX:P
   OKE 752,CB:RETURN
KY 960 C=USR(ADR(RL4#),ADR(O$)-C1+XX,39-X
   X,6)
TV 970 IF C THEN RESTORE 8100+XX+C:READ T
   $:? T$:XX=XX+C:IF XX<RNN THEN GOTO 960
LN 980 IF  NOT XX THEN ? "Nothing of inte
   rest."
NK 990 ? :RETURN
SP 1100 REM INVENTORY
BR 1200 REM INVENTORY
RF 1210 ? "You are carrying the following
   :":XX=CB:G=CB:GOSUB OBJ:RETURN
PV 1300 REM LOAD
XR 1310 IF R<>17 THEN 1350
QC 1320 IF ASC(O$(N,N))<>CL THEN ? "It's
   not here.":RETURN
KX 1330 IF  NOT INN THEN ? NR$:RETURN
RQ 1335 IF RSC(O$(INN,INR)) THEN ? YD$:RE
   TURN
DQ 1337 IF INR<>10 RNO INN<>19 THER ? "Do
   n't be silly.":RETURN
CH 1340 ? "The cannon is loaded.":O$(INN,
   INR)=CHR$(99):ST$(13,13)=CHR$(RSC(ST$(
   13,13))+C1):RO8J=NO8J-C1:NETURN
NZ 1400 NER HEAD
MF 1410 GOSUB NCHECK:I=RSC(O$(N,N)):IF  N
   OT Z OR I<>CL THEN ? NH$:RETURN
XB 1420 IF N<>31 AND N<>25 THEN ? "No goo
   d.":RETURN
OR 1430 IF N=31 THEN ? "The last few page
   s are sieezng, but itdescribee the las
   t voyage.":RETUNN
CU 1440 ? "It rambles on about albatrosse
   s,      curses, and something about so
   me bad  navigation."
NO 1450 ? "Old Jack Tar must have gone a
   little  crazy over the years, judging
   by his  diary.":RETURN
DI 1500 NEM WHISTLE
CN 1510 IF CL(>5 AND CL(>6 THEN ? "You ar
   e out of tune !":RETUNN
TQ 1515 IF ST$(4,4)="8" THEN ? "That has
   no effect now.":RETURN
OV 1520 O$(14,14)=CHR$(99):? "CCHRARSOHH
   !":ST$(4,4)="0"
UW 1530 ? "The icicle resonates and smash
   es. The roof caves in."
NG 1540 IF CL=5 THEN ? "You are swept int
   o the next room.":NL=4:GOSUB NENRR:CL$
   (1,1)=CHR$(0):? "You're trapped.":NETU
   RR
UE 1550 ? "There is a gap in the ceiling
   but the way back is blocked !":CL$(2,2
   )=CHR$(CB):CL$(5,5)=CHR$(7)
CM 1560 JACK=2:GOSUB 8CR:RETURN
LR 1600 REN LIGHT
XB 1610 IF N=17 THEN 1700
KR 1620 SOSUB NCHECK:I=ASC(O$(N,R)):IF Z(
   >0 AND I(>CL THEN ? NH$:RETURN
LO 1630 IF  NOT INN THEN ? NN$:RETURN
WF 1640 IF ASC(O$(INR,INR)) TNEN ? YD$:RE
   TURN
GV 1650 IF INR(>20 THEN ? "It won't work.
   ":RETURN
BY 1660 IF N(>22 RND R(>23 THEN ? "It won
   't light.":RETURN
EQ 1670 IF CL=17 THER ? "The flames attra
   ct the attention of a ship. It pulls t
   owards you.":GOSUB BCR:GOTO RIN
AX 1680 ? "It lights and burns away.":IF
    NOT ASC(O$(N,N)) THER NO8J=NO8J-C1
NX 1690 O$(N,N)=CNR$(99):RETURN
TO 1700 RER FIRE
EE 1705 GOSUB NCHECK:IF ASC(O$(N,R))(>CL
   THEN ? NH$:RETURN
VQ 1710 IF R(>17 THER ? "Silly idea.":RET
   URR
LT 1715 IF  NOT IRN THER ? NN$:NETURN
XE 1718 IF ASC(O$(INN,IRN)) THEN ? YD$:RE
   TURN
BB 1720 IF IRN(>20 THEN ? "It won't work.
   ":RETURN
MK 1725 IF ST$(13,13)(>"C" THEN ? "Booooe
   ! It wasn't loaded and exploded.":O$(1
   7,17)=CHR$(99):RETURN
BH 1730 IF CL=14 THEN ? "Booooee ! The wh
   ole front end of the  ship has been bl
   own away.":CL$(C1,C1)=CHR$(16):JRCK=4
VX 1735 IF CL=14 THER ? "You hear the sou
   nd of splintering woodfrom behind.":GO
   SUB 8CR
JF 1740 ? "The cannon has exploded. You e
   re      lucky not to be hit.":O$(17,17
   )=CHN$(99):RETURN
VH 1800 REN SEARCH
JR 1810 IF  NOT N THER N=1RN
NO 1820 SOSUB NCHECK:Z=ASC(O$(N,R)):IF  N
   OT I OR Z<>CL THEN ? NH$:RETURN
BO 1830 IF N=30 AND ST$(8,8)="R" THER ST$
   (8,8)="8":? "Aha .":O$(18,18)=CHR$(CL)
   :XX=0:8=CL:GOSUB OB3:RETURN
HH 1840 IF N=33 RND ST$(9,9)="A" THEN ST$
   (9,9)="8":O$(11,11)=CHR$(CL):? "Aha!":
   XX=0:O=CL:GOSUB OBJ:RETURN
BN 1850 ? "You find nothing here.":NETURN
FG 1900 REN UNLOCK
OR 1910 SOSUB NCHECK:IF RSC(O$(N,N))<>CL
   THEN ? RN$:RETURN
LG 1920 IF  NOT INN THER ? RN$:RETURN
EM 1930 IF IRN(>34 THEN ? "R key would be
    better.":RETURN
XE 1940 IF N(>C1 THER ? "Strange behaviou
   r.":RETURR
KE 1950 IF ST$(18,18)="C" THER ? "It's al
   ready unlocked.":RETURR
LI 1960 IF ST$(18,18)="R" THEN ST$(18,18)
   ="8":? "It's unlocked now.":IF JACK=3
   THER JACK=4
BN 1970 RETURN
UB 2000 REN LOCK
OR 2010 GOSUB NCHECK:IF RSC(O$(N,N))<>CL
   THEN ? RN$:RETURN
KP 2020 IF  NOT INN THEN ? NR$:RETURR
DV 2030 IF IRN(>34 THER ? "A key would be
    better.":RETURR
MN 2040 IF N(>C1 THEN ? "Strange behaviou
   r.":NETURN
IR 2050 IF ST$(18,18)="C" THEN ? "Close i
   t first !":RETURN
JF 2060 IF ST$(18,18)="A" THER ? "It's al
   ready locked.":RETUNN
UB 2070 ? "It's now locked.":ST$(18,18)="
   A":IF JACK=2.5 THER JACK=3:GOSUB BCR:R
   ETURN
RY 2080 RETURN
KC 2100 RER CLOSE
DC 2110 GOSUB RCHECK:IF ASC(O$(R,N))<>CL
   THEN ? RN$:RETURN
OT 2115 IF N=4 THER ? "You've broken it -
    it won't close now.":RETURR
LL 2120 IF N(>C1 THER ? "Nice try.":RETUR
   N
PL 2130 IF ST$(18,18)="C" TNEN ST$(18,18)
   ="8":? "Closed.":RETURN
OO 2140 ? "It's already closed.":RETURN
YF 2200 REN OPEN
DE 2210 GOSUB NCHECK:IF RSC(O$(N,N))<>CL
   THEN ? NH$:RETURN
HP 2220 IF R<>C1 RNO R<>6 AND N<>4 THEN ?
    "Strange idea.":RETURN
NR 2222 IF N=4 AND ST$(12,12)="A" THEN ?
   "Aha.":O$(20,20)=CHR$(27):XX=CB:S=27:8
   OSUB OBJ:ST$(12,12)="8":NETURNN
OA 2223 IF N=4 THER ? "It's already open.
   ":RETURN
LE 2225 IF N=6 THEN ? "You looked, and th
   e cupboard was bare.":NETURN
VF 2230 IF ST$(18,18)="A" TNEN ? "It's lo
   cked.":RETURN
IE 2240 IF ST$(18,18)="8" THEN ? "It is o
   ee open.":ST$(18,18)="C":RETURN
RI 2250 ? "It is already open.":RETURN
EG 2300 NEN PUSH
BB 2310 GOSUB RCHECK:IF ASC(O$(R,R))<>CL
   THER ? RN$:RETURR
OP 2320 ? "That has very little effect.":
   RETURN
BC 2400 REN QUIT
XJ 2410 ? "Ok.":GOTO 710
RG 2500 REM HIT
RP 2600 REM KICK
FE 2610 GOSUB NCHECK
UO 2620 IF ASC(O$(N,N))<>CL THER ? RN$:RE
   TURN
RN 2625 IF N=13 RND ST$(3,3)="8" THEN SOS
   UB 8CR
OY 2630 IF N=13 AND ST$(3,3)="8" THEN ? "
   The ice dinner-plates and seashes.":CL
   $(1,1)=CHR$(5):ST$(3,3)="C":RETURN
MO 2640 IF N=7 O$ R=8 THEN ? "They dodge
```

```
   out of the way.":RETURN
RM 2650 IF N=14 THEN ? "It vibrates a bit
   - It's strong !":RETURN
OT 2660 ? "Can't waste your time.":RETURN
YZ 2700 REM STAB
JK 2800 REM FIGHT
NN 2900 REM KILL
DS 2910 GOSUB NCHECK:IF ASC(D$(N,N))<>CL
   THEN ? NH$:RETURN
FN 2920 IF N<>7 AND N<>8 THEN ? "What a w
   aste of time.":RETURN
LK 2930 IF  NOT INN THEN ? NH$:RETURN
NN 2940 IF ASC(D$(INN,INN))  THEN ? YD$:RE
   TURN
TN 2950 IF iNN<>26 THEN ? "You can't do t
   hat with one of those.":RETURN
DX 2960 IF N=7 THEN ? "They dodge out of
   the way.":RETURN
UF 2970 ? "Jack falls back dying. 'The cu
   rse is  over, Captain Ahab.' he breath
   es. He  dissolves from your sight."
VQ 2980 GOSUB SCH:D$(0,0)=CHR$(99):JNCK=6
   :RETURN
HY 3000 REM TAKE
KJ 3100 REM GET
EV 3110 GOSUB NCHECK
EG 3120 IF  NOT (ASC(D$(N,N)))  THEN ? "Yo
   u already have that !":RETURN
UI 3130 IF ASC(D$(N,N))<>CL THEN ? NH$:AE
   TURN
YT 3140 IF N<15 THEN ? "You can't have th
   at !":AETURN
FQ 3142 IF N=36 OR N=15 THEN GOSUB SCR
NF 3145 IF N=33 AND ST$(9,9)="A" THEN ? "
   It won't budge !":RETURN
PN 3146 IF N=38 AND ST$(8,8)="A" THEN ? "
   They are quite heavy."
CJ 3147 IF (N=16 OR N=15) AND CL=22 AND S
   T$(11,11)="A" THEN ? "The rope bite yo
   ur hand - you can't  have it.":RETURN
LE 3148 IF N=27 THEN ST$(11,11)="A"
UH 3150 IF NOBJ<6 THEN ? "Ok.":D$(N,N)=CH
   R$(0):NOBJ=NOBJ+1:RETURN
ON 3160 ? "You're carrying too much !":RE
   TURN
ZU 3200 REM LOOK
SH 3300 REM EXAMINE
JJ 3305 IF  NOT N THEN N=INN
PX 3306 IF  NOT N THEN ? "Ok.":NL=CL:GOSU
   B MENAN:RETURN
LC 3310 Z=ASC(D$(N,N)):IF Z AND Z<>CL THE
   N ? NH$:AETURR
RF 3315 IF N=13 THEN ? "You see a section
   of thinner ice.":ST$(3,3)="B":AETURN
FA 3320 IF N=14 THEN ? "It's lovely.":RET
   URN
JO 3325 IF N=3 AND ST$(4,4)="A" THEN ? "A
   he !":D$(4,4)=CHR$(27):XX=C0:B=CL:GOSU
   B OBJ:ST$(4,4)="B":RETURN
NJ 3330 IF N=4 THEN ? "It's an antique. L
   ooks like it's      hinged.":RETURN
DE 3335 IF N=17 THEN ? "This one looks in
   better nick.":RETURN
KZ 3360 IF N<>2 THEN ? "You see nothing s
   pecial.":RETURN
QC 3365 IF ST$(5,5)="B" THEN ? "Nothing s
   pe.":RETURN
QG 3370 ? "You notice a small key in her
   mouth ":D$(34,34)=CHN$(36):D$(35,35)=
   CHN$(36):GOSUB OBJ:ST$(5,5)="B":RETURN
WA 3400 REM CHOP
OZ 3500 REM SAW
YQ 3510 IF N<>37 THEN ? "You can't saw th
   at.":RETURN
HN 3520 IF CL<>19 AND CL<>20 THEN ? "You
   can't get the planking from here.":RET
   URN
LO 3530 IF  NOT INN THEN ? NH$:RETURN
NF 3540 IF ASC(D$(INN,INN))  THEN ? YD$:RE
   TURN
UJ 3550 IF INN<>15 THEN ? "Doesn't work."
   :RETURN
ZV 3560 ? "Ok. You've got some planking."
   :IF CL=19 THEN ST$(14,14)="B"
UR 3570 D$(37,37)=CHR$(CL):? "The saw has
   unfortunately broken.":D$(15,15)=CHR$
   (99):NOBJ=NOBJ-1:RETURN
HO 3600 REM BUILD
NC 3610 IF N<>24 THEN ? "You want to buil
   d what ?":RETURN
AB 3630 IF ASC(D$(37,37))  THEN ? "You nee
   d some planking.":NETURN
ZI 3640 IF ASC(D$(38,38))  THEN ? "You nee
   d some nails.":AETURN
BY 3650 IF ASC(D$(39,39))  THEN ? "You nee
   d a hammer.":RETURN
JZ 3660 ? "With a lot of difficulty, you
   build a small raft. Well done.":D$(24,
   24)=CHR$(CL):D$(38,38)=CHR$(99)
OO 3670 D$(37,37)=CHN$(99):NOBJ=NOBJ-2:GO
   SUB SCR:AETURN
IE 3700 REM NEIGH
DP 3710 GOSUB NCHECK:IF ASC(D$(N,N))<>CL
   THER ? NH$:NETURN
HB 3720 IF N<>11 THEN ? "Feels heavy.":NE
   TURN
SR 3730 ? "You just manage to throw it ov
   er the  side.":D$(11,11)=CHR$(99):GOSU
   B OBJ:ST$(7,7)="B":GOSUB SCA
GQ 3740 ? "Looks like you could get down
   it.":CL$(6,6)=CHR$(15):RETURN
YF 3800 AEM LAUNCH
DM 3810 GOSUB NCHECK:Z=ASC(D$(N,N)):IF Z
   AND Z<>CL THEN ? NH$:NETURN
GG 3820 IF N<>24 THEN ? "You don't make s
   ense.":RETURN
QW 3830 IF CL<>16 THEN ? "You can't launc
   h from here.":NETURN
ZZ 3840 NL=17:GOSUB NENNN
MC 3850 IF ST$(14,14)="A" THEN AETURN
YM 3860 ? "You made it with rotten wood.
   It sinksalmost. Oh dear, you've drowned
   ed.":GOTO LOSE
BA 3900 AEN CHOP
FL 3910 GOSUB NCHECK
FR 3920 IF ASC(D$(N,N)) THEN ? YO$:RETURN
YY 3930 NOBJ=NOBJ-C1:? "Ok.":D$(N,N)=CHR$
   (CL)
QH 3935 IF N=36 OR N=15 THEN SCORE=SCORE-
   10
GG 3940 IF N=32 AND INN=C1 AND P=C1 THEN
   ST$(C1,C1)="C":RETURN
LT 3950 IF N=32 THEN ST$(C1,C1)="O":RETUN
   N
NO 3960 IF N=27 AND CL=22 AND INN=7 THEN
   ST$(11,11)="B":? "The rope are trapped
   .":D$(27,27)=CHR$(22):NETURN
BO 3970 NETURN
NG 4000 REM NEAN
EU 4010 GOSUB NCHECK
LA 4020 Z=ASC(D$(N,N)):iF Z AND Z<>CL THE
   N ? NH$:NETURN
CV 4030 IF N=35 THEN ? "It fits well unde
   rneath your shirt.":ST$(2,2)="O":RETUN
   N
BN 4040 IF N=21 OR N=23 THEN ? "Doesn't s
   uit you.":RETURN
VN 4050 ? "What a strange thing to wear !
   ":NETURN
CU 4100 REM CLIMB
IP 4110 iF  NOT N THEN N=INN
MV 4115 IF N=17 AND CL=24 THEN NL=18:GOSU
   B NEWRN:RETURN
YG 4117 IF N=9 AND CL=8 THEN ? "But you a
   re already on it.":RETURN
KO 4120 GOSUB NCHECK:IF N=12 OR N=18 THEN
   4160
VA 4125 IF ASC(D$(N,N))<>CL THEN ? NH$:RE
   TURR
DQ 4130 IF N=13 OR N=14 THEN ? "It's too
   slippery.":NETUNN
GP 4145 IF N=29 THEN ? "Ok.":NL=32:BOSUB
   NENRN:AETURN
GL 4155 IF N=9 THEN NL=8:GOSUB NENNN:NETU
   RN
JH 4157 ? "Climbing that won't achieve su
   ch.":RETUNN
ZH 4160 IF N=12 THEN 4188
LM 4165 IF CL=8 THEN NL=9:GOSUB HENRR:NET
   URN
NG 4166 IF CL=9 THEN RETURN
QY 4167 IF CL=10 THEN NL=25:BOSUB NENRN:N
   ETURN
DV 4170 IF CL=25 THEN NL=18:BOSUB HENRN:N
   ETURN
ZZ 4172 IF CL=15 THEN NL=14:BOSUB NENRN:N
   ETURN
OC 4174 IF CL=14 AND ST$(7,7)="B" THEN NL
   =15:GOSUB NENNN:NETUNN
YB 4176 IF CL=14 THEN ? "You'll fall !":R
   ETURN
PL 4178 ? NH$:RETURN
FV 4180 IF CL=27 THER NL=28:GOSUB NENRN:N
   ETURN
BJ 4182 IF CL=34 THEN NL=33:GOSUB HENRN:R
   ETURN
EX 4184 IF CL=13 THEN NL=29:GOSUB NENRN:N
   ETUNN
```

SOFTWARE EXPRESS

514-516 Alum Rock Road, Alum Rock, Birmingham Tel.: 021-328 3585

STOP PRESS STOP PRESS

WE ARE VERY PLEASED TO ANNOUNCE OUR ACQUISITION OF THE MIDLAND ATARI CENTRE RIGHT IN THE HEART OF BIRMINGHAM AT 212 BROAD STREET (TEL: 021-643 9100)

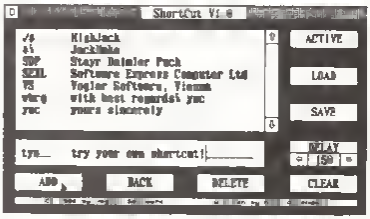
OUR MAIN OPERATION CENTRE WILL REMAIN AT ALUM ROCK ROAD AND WE SHALL CONTINUE TO RETAIL FROM THIS LOCATION TO CATER FOR OUR CUSTOMERS CONVENIENCE.

VOGLER Software

ShortCut

Save yourself time and trouble when typing. ShortCut enables you to define your own easy to remember "shortcuts" using standard keyboard characters.

£29.95



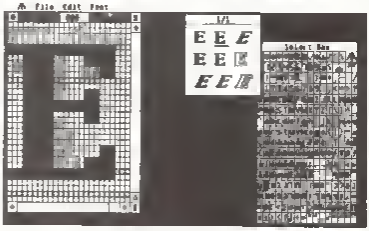
JackFont
JackSpell
JackMake
HighJack
Shortcut

JackFont

A font editor for system fonts and VDI metafile fonts up to 72 point.

JACKFONT is compatible with FLEET STREET PUBLISHER and any program supporting GDOS 1.1

£49.95



GAUNTLET

£24.95

ATARI ST

514-516 Alum Rock Road,
Alum Rock, Birmingham

HOTLINE
021-328 3585

Dealer and Overseas Enquiries Welcome

```
OG 4186 IF CL=29 THER NL=30:GOSUB NEWRR:R
   ETURN
YL 4188 IF CL=30 THER NL=31:GOSUB NEWRM:A
   ETURA
DL 4190 ? RN$;RETURA
ZB 4200 REM FORAARD
ZO 4300 RER FORAARD
1T 4400 RER AFT
IV 4500 RER AFT
ND 4600 RER PORT
N9 4700 RER PORT
PY 4800 AEM STARBOAAD
DA 4900 REN STARBOARD
HG 5000 REM UP
AS 5100 AEA UP
QZ 5200 REN DOAA
AE 5300 AL=ASC(CL$(22-INT(V/2)))
BL 5305 IF AL=26 AND JACK=2 THER ? "Sudde
   nly Jack appears, e frown on hie face.
    He will not let you pass.";RETURR
PZ 5310 IF AL()33 THEN GOSUB AEAAA:AETURR
ZT 5320 IF ASC(D$(29,29))=31 THEN Q$(29,2
   9)=CHR$(99):? "Just in time ! The rigg
   ing collapses !":GOSUB NEARA:RETURR
JS 8001 DATA In the sea
SC 8002 DATA Under the sea
BH 8003 DATA Ice cave
BA 8004 DATA Ice wall
DV 8005 DATA Ice tunnel
DZ 8006 DATA Ice tunnel
SP 8007 DATA Cavern
UE 8008 DATA On the rudder
KA 8009 DATA Captain's bunk
RD 8010 DATA Corridor
PI 8011 DATA Chart room
AD 8012 DATA Store room
CA 8013 DATA Deck
FS 8014 DATA Crew e bunks
KJ 8015 DATA Anchor chain
LY 8016 DATA Ice platform
MV 8017 DATA On the sea
PZ 8018 DATA On the cannon
SY 8019 DATA Corridor
RR 8020 DATA Corridor
FQ 8021 DATA Galley
AQ 8022 DATA Store room
OQ 8023 DATA Cannon gallery
GH 8024 DATA Cannon gallery
AS 8025 DATA Store room
GH 8026 DATA Jack's lair
EJ 8027 DATA Poop deck
FV 8028 DATA Mizzen east
NL 8029 DATA Main east
GJ 8030 DATA High on main east
UP 8031 DATA Crow s nest
ZO 8032 DATA Rigging
PU 8033 DATA Fore east
VZ 8034 DATA Forecastle
TU 8035 DATA Bow
AN 8036 DATA On the bowsprit
EA 8101 DATA large box
YV 8102 DATA ornate figurehead
1X 8103 DATA binnacle
JD 8104 DATA magnetic compass
BZ 8105 DATA barrel
TY 8106 DATA cupboard
UN 8107 DATA rats
YU 8108 DATA Jack Tar
RT 8109 DATA rudder
NS 8110 DATA tiny porthole
FI 8111 DATA heavy anchor
SO 8112 DATA sext
SO 8113 DATA cold ice
XZ 8114 DATA icicle
NI 8115 DATA foodsae
LO 8116 DATA food
AS 8117 DATA old cannon
GP 8118 DATA ball
AO 8119 DATA gunpowdwr
AT 8120 DATA fiist
JF 8121 DATA deep curtain
RV 8122 DATA small rug
AC 8123 DATA sailcloth
OE 8124 DATA raft
AI 8125 DATA diary
NA 8126 DATA evil knife
LI 8127 DATA net
RD 8128 DATA sextant
ZT 8129 DATA rigging
ZP 8130 DATA blanket
NU 8131 DATA log
NV 8132 DATA lucky doubloon
ON 8133 DATA hawser
LT 8134 DATA small key
X8 8135 DATA rusty breastplate
TE 8136 DATA treasure chest
AY 8137 DATA planking
YT 8138 DATA nails
SN 8139 DATA piece of iron
DS 8202 DATA You are bobbing on the freez
   ing cold  swell. You are drifting near
    a large  bluish iceberg.
UN 8203 DATA It is floating very low in t
   he water.
JY 8204 DATA It's gloomy and cold underne
   ath the   surface. Weird reflections p
   lay in thewaters in front of you.
JY 8205 DATA Your breath is running out.
   There is  something ahead !
IN 8206 DATA You are inside an ice-cave e
   ithin the iceberg - but still underwat
   er. Your  muscles are freezing up.
DL 8207 DATA Looks like you can escape up
   wards.
BZ 8208 DATA You are now on an ice-ledge
   by the    side of the freezing wall yo
   u have    just swum through.
NG 8209 DATA Walls of ice surround you.
XK 8210 DATA The tunnel is dominated by a
    beautiful slender icicle that stretche
   s from    floor to ceiling.
YG 8211 DATA It glows from a pale light f
   rom above.
IN 8212 DATA You are now on the other sid
   e of the  slender icicle. The corridor
   ends      abruptly here.
DF 8213 DATA
DV 8214 DATA You are in a large open cave
   rn. Your  south drops as you see a hug
   e old     galleon embedded
SM 8215 DATA deeply in the ice. Down here
    by the   rudder it seems lighter towa
   rds the   front of the galleon.
RY 8216 DATA You are balanced precariousl
   y on the  icy rudder. There is a small
    porthole just within reach.
DV 8217 DATA
HT 8218 DATA You are now inside the Capta
   in's      quarters. It's a bit dusty i
   n here.
ED 8219 DATA
YS 8220 DATA This passageway has doors an
   d hatches everywhere.
DA 8221 DATA
TF 8222 DATA This is the old chart room -
    ornately decorated.
01 8223 DATA
LB 8224 DATA This is an old store place -
    smelling of rotting wood.
DA 8225 DATA
MS 8226 DATA This is the deck area with d
   ebris     everywhere. It seems to get
   lighter   towards the forecastle.
LR 8227 DATA The way up to the forecastle
    is barredby a huge twisted ice-fall.Y
   ou are    underneath the main mast.
KL 8228 DATA Here are the crew s sleeping
    quarters situated at the front of the
    ship.
CV 8229 DATA The wood is very rotten here
    and there a small porthole nearby.
CC 8230 DATA You are dangling on the anch
   or chain  with wooden walls in front a
   nd ice    walls behind.
NG 8231 DATA Below is a small porthole an
   d above   is the forecastle deck.
CA 8232 DATA You are standing on the ice
   platform  blasted out from the iceberg
   . The sea laps against the ice.
DL 8233 DATA
GB 8234 DATA You are on the sea. It is qu
   ite near  to a shipping lane judging b
   y all the flotsam.
DT 8235 DATA
CO 8236 DATA You are hanging from the end
    of the   cannon and below you can see
   s small   porthole.
EB 8237 DATA
BN 8238 DATA This is a rotten section of
   corridor -it could collapse anytime. T
   here is   some planking here.
EJ 8239 DATA
AI 8240 DATA This section has lasted much
    better   than the corridor in front.
   There is  some planking here.
DG 8241 DATA
1F 8242 DATA This is the old cooking area
    - there  are dried foodstains everywh
   ere.
```

```
DD 8243 DATA
KY 8244 DATA This is a very pongent store
   room.     There are some rats with hug
   e glowing eyes in the corner.
DA 8245 DATA
AZ 8246 DATA This is the cannon deck. All
    the guns seem to be suffering from ag
   e with    rust all over them.
EE 8247 DATA
YI 8248 DATA There is only one wrecked ca
   nnon left here and it is rigidly fixed
    right at the end of it's gun-rail.
EA 8249 DATA
FA 8250 DATA This is a dusty old room fil
   led with  hammar and sail-cloth. there
    are no   exits here except the portho
   le.
BJ 8251 DATA
TZ 8252 DATA This is old Jack Tar's hemu.
    No wonderhe didn't want you here - it
   s filled with all sorts of oddments.
DN 8253 DATA
CA 8254 DATA You are standing on the poop
    deck     beneath the rear mast.
DZ 8255 DATA
TY 8256 DATA You have scaled the mizzen m
   ast. It isdefinetely quite high up.
EA 8257 DATA
AM 8258 DATA This is a very tall mast. Yo
   u are onlya third of the way up.
EP 8259 DATA
FK 8260 DATA You are near the top of the
   main mast and can see the whole galleo
   n far     below. Your arms are tired.
DM 8261 DATA
MZ 8262 DATA You are now on the crow's ne
   st lookingdown on the decks with their
    tattered sails.
KD 8263 DATA The sails have all fallen bu
   t some of the frayed rigging remains i
   n place.
BA 8264 DATA You are hanging on a single
    line     between two masts. It won't
    hold your weight for much longer.
EC 8265 DATA
FD 8266 DATA You are now on the fore-mast
    above theforecastle and ice-fall.
EK 8267 DATA
RV 8268 DATA You ars on the deck of the f
   orecastle.The jumbled icefall blocks t
   he way    down to the main deck.
EB 8269 DATA
LV 8270 DATA You are at the very front of
    the dsck.Ahead is the bowsprit.
BP 8271 DATA
BF 8272 DATA You are astride the bowsprit
    in front of the ship. The bows are ve
   ll lit.
BX 8273 DATA
YD 8301 DATA 1,1,1,1,0,2
XA 8302 DATA 3,2,2,2,1,2
UA 8303 DATA 0,2,0,0,4,0
SR 8304 DATA 0,0,0,0,0,3
UY 8305 DATA 6,4,0,0,0,0
TB 8306 DATA 0,5,0,0,0,0
VD 8307 DATA 0,0,0,0,0,6
AA 8308 DATA 0,0,0,0,0,7
JP 8309 DATA 10,8,0,0,0,0
AA 8310 DATA 13,9,12,11,0,20
EG 8311 DATA 0,0,10,0,0,0
PA 8312 DATA 0,0,0,10,0,0
AD 8313 DATA 14,10,0,0,27,19
GH 8314 DATA 0,13,0,0,0,0
FH 8315 DATA 0,0,0,0,35,14
NI 8316 DATA 0,14,0,0,0,0
AJ 8317 DATA 17,17,17,17,0,0
JI 8318 DATA 0,0,0,0,24,0
XN 8319 DATA 0,20,23,24,13,26
GJ 8320 DATA 19,21,23,24,10,0
AU 8321 DATA 20,22,0,0,0,0
FI 8322 DATA 21,0,0,0,0,0
LO 8323 DATA 0,0,0,19,0,0
KT 8324 DATA 0,0,19,0,0,0
GD 8325 DATA 0,0,0,0,0,0
MX 8326 DATA 0,0,0,0,19,0
SZ 8327 DATA 0,0,0,0,28,13
NG 8328 DATA 0,0,0,0,0,27
BA 8329 DATA 0,0,0,0,30,13
NA 8330 DATA 0,0,0,0,31,29
HE 8331 DATA 0,0,0,0,0,30
YV 8332 DATA 33,31,0,0,0,0
KO 8333 DATA 0,0,0,0,0,34
DO 8334 DATA 35,0,0,0,33,0
CK 8335 DATA 36,34,0,0,0,0
JC 8336 DATA 0,35,0,0,0,0
KM 9000 DIM V$(215),M$(195),P$(30),U$(5),
   K$(15),IN$(60),AL1$(67),NL2$(30),ML3$(
   30),AL4$(30),A$(216),CL$(6),T$(120)
LN 9005 C0=0:C1=1
WJ 9010 AESTONE 9040:FON A=C1 TO 66:AEAD
    D:NL1$(A,A)=CHR$(D):AEXT A:FON A=C1 TO
    30:NEAD D:AL2$(A,A)=CAN$(D):NEXT A
FY 9020 FOA A=C1 TO 30:AEAO D:ML3$(A,A)=C
   AA$(D):NEXT A:NL1=AOA(AL1$):AL2=AOR(NL
   2$):AL3=AOR(AL3$):G0=100
QJ 9030 FON A=C1 TO 30:AEAD O:ML4$(A,A)=C
   NA$(D):AEXT A:NL4=AON(NL4$)
ZJ 9040 DATA 104,104,133,209,104,133,208,
   104,133,213,104,133,212,104,133,211,10
   4,133,210,160,0,177,208,209,210
EV 9050 DATA 240,31,165,208,24,105,5,133,
   208,144,2,230,209,165,212,56,233,1,133
   ,212,165,213,233,0,133
XQ 9060 DATA 213,208,222,165,212,208,218,
   96,208,192,5,144,214,132,203,96,104,10
   4,133,211,104,133,210,104,104
NF 9070 DATA 170,104,104,160,0,132,212,13
   2,213,209,210,240,5,200,202,208,248,96
   ,132,212,96,104,104,133,211
NU 9080 DATA 104,133,210,104,104,170,104,
   104,160,0,132,212,132,213,209,210,208,
   5,200,202,208,248,96,132,212,96
YA 9090 DATA 104,104,133,204,104,133,203,
   104,104,170,104,104,160,0,132,213,200,
   209,203,240,6,200,202,209,246
EG 9100 DATA 160,0,132,212,96
PL 9200 V$(C1,80)="F    FOAAAA    AFT  P
      POAT B    GTARBU    UP    D    DOWN
   CLIABAEAA 9AOP LAUAC"
BA 9201 V$(81,160)="AEIGHBUILDSAA  CHOP E
   XAMILOOK GET  TAKE KILL FIGHTSTAB KICK
    HIT  QUIT PUSH OPEA "
TA 9202 V$(161,215)="CLOSELOCK UNLOCSEAAC
   FINE LIGHTWHISTAEAD LOAD TAVENI     "
FF 9210 A$(C1,80)="INOM NAILSPLANKTAEASON
   EASKEY  HAAGEOOUBLLDS  BLAAKA1BGISEXTA
   NET  KN1FEDIAAYAAFT "
VG 9211 N$(81,160)="SAILCNUS  CUNTAFL1ATS
   UAPODALL CAAAOFOOO FDOBSICICL1CE  AAST
    ANCHOPORTHAUDDEJACK "
AP 9212 A$(161,195)="AATS CUPBOBAWAECOAPA
   DIANAFIGUAQOX  "
AB 9220 P$(C1,30)="THROUMITH ON   FAOR IN
      AT   "
FE 9230 U$(C1,5)="TAE  "
XI 9240 AV=43:NNA=39:NP=5
1X 9250 AEM
EN 9259 LET AEMAA=980:DESC=930:AAAE=940:D
   BJ=960:NCNECK=890
AV 9260 AESTOAE 8300:FOA A=C1 TO 216:NEAD
    D:M$(A,A)=CHN$(D):NEXT A:CL$(C1,6)=N$
   (C1,6):CL=C1:NL=C1
HC 9300 DIM O$(39),GT$(20)
MT 9310 RESTOAE 9320:FOA A=C1 TO 39:READ
    D:O$(A,A)=CHR$(D):NEXT A
QC 9320 DATA 13,36,27,99,26,9,22,99,7,99,
   99,99,4,5,22,22,23,99,26,99,9,9,34,99,
   26,26,12,11,31,14,11,99,35,99,99
SU 9330 DATA 25,99,31,28
XG 9350 AD&J=C0:ST$(C1,14)="AAAAAAAAAAAAA
   A":JACK=C1:AIN=0:DOU5=2
TI 9360 AESTOAE 9370:DIA AL5$(11):FOA A=C
   1 TO 11:AEAD D:ML5$(A,A)=CHN$(D):NEXT
   A
YB 9370 DATA 104,169,0,160,39,145,00,136,
   16,251,96
JS 9380 SCORE=C0:POKE 16,64:POKE 53774,64
HI 9400 SOSUB NEWRM:GOSUB NANE:TRAP 230
NL 9410 MIN=720:LOSE=700:SCR=750
LO 9420 NESTONE 9420:FOR A=0 TO 10:NEAO D
   :POKE 1536+A,D:NEXT A:DATA 72,169,144,
   141,10,212,141,24,208,104,64
DA 9430 POKE 512,0:POKE 513,6:DL=PEEK(560
   )+256*PEEK(561):POKE DL+3,PEEK(DL+3)+1
   20:POKE 710,4:POKE 54286,192
TF 9500 DIA AH$(14),AA$(10),YD$(20)
1H 9510 0A$="It's not here.":0A$="Alth wh
   at ?":YB$="You don't have that."
WZ 9999 GOTO 140
```

Lightspeed C on the 130XE

From Clearstar Softechnologies
Review by Walter W. Jones
Reprinted from Current Notes

To begin a comment about the Atari 600XL 800XL and 130XE series of computers (hereafter referred to as the 130) is appropriate. The lack of a commercial high level language has, obviously, prohibited its extensive use on Atari's 8 bit line But the easiest way to get new programs is to port them from another system and fix the few machine dependent pieces The cost of porting a program to a new computer is only about 10 per cent of the original development cost, assuming that a comparable language exists on the target computer Basic is too slow to do serious work and not very compatible anyhow Action!, although a vast improvement over Basic, is not directly compatible with any other language

Two features which are indispensible for a development language are compact coding and linkable modules The former is necessary so that writing all of the necessary but simple routines does not take an inordinate amount of time, while the resultant object code is reasonably fast. The second feature is the ability to code and debug modules This is the only way to make them quickly and without having to reinvent the necessary low level stuff each time For this reason, 'programmer friendly' languages have libraries and means to combine modules They emphasise the concepts of local and hidden over global and public Otherwise the scaffolding necessary to debug the code is impossible to build, and the time for translation becomes tedious

The first requirement has sent all serious development over to Action! However, the second requirement suggests that little will be developed for the 130 Until now that is

Clearstar Softechnologies has brought a product to the market which solves both of these problems in a very nice way There are still some difficulties and shortcomings, but the primary impediment to software development and portability on these 'low level' computers is removed The compiler is for a C Language C is a popular language for many systems It was the first language to be used widely in systems development work which was a significant step above assembly language C was a step up in that it was possible to write compact code which was still fast Many dialects of C have developed Lightspeed C incorporates one of the more common variations and is a subset of the standard Ritchie and Kernigan specification

What all this means is that programs developed on the ST can be ported to the 130 and vice versa Obviously there will be some differences, especially in graphics But much of the translation and recoding problems disappear To prove the point, I wrote a simple fractal generator in C and ran it on my 130XE, my IBM AT and an Apollo workstation In each case I wrote each module separately compiled tested and stored it on a disk file With the allowance for differences in graphics, all tests performed the same way The 130 is slower than the Apollo, on the other hand the latter cost 100 times as much

The Lightspeed C development system comes as two disks, or four sides of our single sided floppy disks Effectively there are only two sides required and these are contained on sides one and two of the first disk What the authors have done to further enhance development is to include many of the special functions within their own operating system To aid use and distribution the second disk contains alternate compilers and linkers which do not depend on their specialised DOS being present Libraries are provided for most of these functions The combination is a very nice touch and should be appreciated by anyone doing development for the Atari 130XE

The mechanics of using the system will be familiar to anyone who has done development work on a large system There is a full screen editor, a compiler and a linker The editor serves both for writing the program and for looking at errors during compile This is where I sorely miss a hard disk on my 130 There is a fair bit of code swapping which takes place Once again to the credit of Clearstar, all the necessary modules fit on a single disk, or RAMdisk, with room left for pieces of code Actually it is not hard to have the source on one disk, say a floppy and the compiler, linker and editor in RAMdisk Load times for the compiler, linker and editor then become miniscule To facilitate such usage, a batch file processor is provided which automates the setup procedure

There is another feature well worth mentioning A library of Action! entry points, including the floating point routines is part of the distribution This makes it possible to start with Action! and convert to C Names are unique, or perform equivalent tasks For anyone considering a move from the 8 bit to the 16 bit world, this feature is outstanding Naturally, it is possible to reverse the process and convert to Action! Since compiled Action! code is somewhat faster than the equivalent compiled C code, this is worthwhile for that last little bit of speed necessary for good simulations This crosswalk between systems provides the continuity lacking in the Atari Corp offerings Finally there are a number of modules to aid in code development and debugging, i.e. for code compaction and optimization Also included are several sample programs in C and Action! just to get you started

As might be expected, the product is not perfect C is not a friendly language It is a compiled language with a very strict syntax This compiler does not improve on that in that the pointer for errors, as well as the error message themselves can be misleading Two additional faults stand out. The first is that the editor is not easy to use in screen mode It is too much like the Basic editor Perhaps it utilizes the hooks in the ROM, but the Action! editor is certainly better Of course, one can use the Action! editor, or even one's favourite word processor But this negates some of the usefulness in the tight coupling between the compiler and the editor The second point is that the floating point routines are difficult to use just as in Action! And there is no floating point type Since other types are defined, it would have been a good place to include them After all, 6 byte floating point is not intrinsically different than a string, or a 2 byte integer The machines still have an 8 bit buss The worst failing is the manual which is atrocious Besides being a hodgepodge of information, the important facts are well hidden

Why then is this such a good product? Well it is a reasonable subset of the standard It makes it possible to port programs to the 130 from other systems so we obtain good value, while the cost to the developer is minimised Also it follows the form of compilers and editors on larger machines and so makes an ideal training tool for anyone wanting to upgrade later I would anticipate that revisions of the Lightspeed system can and will solve the relatively minor problems Lightspeed C retails for $39 95 in the U S A

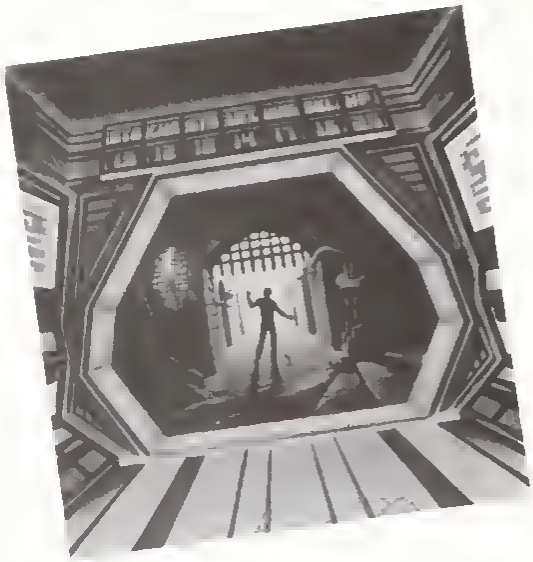
Alternate Reality The Dungeon

From Datasoft/US Gold
Price £19 99 Diskette
Review by Brad Mountjoy

This is the second instalment of the Alternate Reality series, the first was the City, future episodes being the Arena, the Palace, the Wilderness, Revelation and Destiny You don t have to have the City to play this game though You start at the entrance to the Dungeon where your levels of Stamina Strength, Charm, Wisdom, Intelligence Skill, Wealth and Hit Points are randomly selected for your character Keep an eye on your status because you can suffer from the cold or heat, you can be hungry, thirsty, tired, diseased, poisoned, cursed, etc Buying a compass from the Damon & Pythias Shoppe could be a good move When you enter the Dungeon you start with neutral morality you are responsible for your own moral alignment You could become evil or good It's up to you! Evil acts include attacking good or neutral life forms stealing, and tricking any life form even evil ones

An incomplete map of the Dungeon is provided in the 40 page manual marked in a grid for you to complete. The grid is 64 x 64 and can be used for the Level 1 map, there are however deeper levels, the second being 32 x 32, the third 16 x 16 and the final level just 8 x 8 About 4 minutes is equal to an Alternate Reality hour, this can ba an advantage when you need to sleep for 15 hours to replenish your strength There are many different types of doors in the Dungeon some are invisible or require special skills to be opened, some are locked, bolted or enchanted Walls can be strange too, some are one way and some are made of crystal. There are plenty of items or bits of treasure to be found or purchased from the appropriate store Torches, food packets water flasks, clothing, compasses, timepieces, weapons and armour, keys, gems, money, crystals, wands, scrolls, trump cards, eyes and horns to mention but a few

You need a torch to find your way about, or they can be used as secondary weapons (when in an encounter situation you have a primary weapon and a secondary back up) Clothing is important as it could prevent you either freezing or burning to death, it can also have magical properties, i e an Elven Cloak lets you blend into the background or a Crystal Belt adds to your stamina Keys are useful for opening locked doors gems and jewels can be exchanged for money or used as gifts or trade items, money is in gold, silver or



copper There are several different wands each having a different magical purpose, and this is also the case for horns and eyes, i e the Hypnotic Eye bewilders all monsters, the Gold Horn heals your wounds and the Fear Wand makes 'most monsters retreat in terror

There is a massive choice of weapons and armour available and they all have a battle value which is determined by your opponent, your moral alignment and your strength and skill Weapons can also be cursed and enchanted and in addition are good or evil this can cause you problems as you try to slay a harmless pauper with a good sword the sword screams and drains your hit points Ammunition is also needed for some weapons, like a crossbow and your ammo count is shown on the screen

Magic Scrolls and Trump Cards can be used to aid your progress through the Dungeon, there's a remove curse scroll and a wizard's eye scroll which lets you see secret doors, the Fool card increases your luck, the Heirophant card summons a healer, the Temparance card cures drunkenness, etc. There are many potions available from Lucky's Potion Brewery and he will also inform you of the use of any potion you find, well for a fee anyway! Many potions are cures for diseases or antidotes to poisons, some allow you to see in the dark, others increase your hit points during battle by phenomenal amounts or heal your wounds

Places of interest in the Dungeon include Der Rathskeller Bar & Grille and the Retreat ale house Fountains are useful because they provide a good drink and also provide special magic like healing, cleansing diseases, removing fatigue, etc There are bank vaults to rob, but beware of the guards, there's a weapons enchantress who repairs your favourite sword In the Chapel you can pray, hear a sermon or consult a priest, if you are on the dark side you could gain the chance to repent The Dwarven Smithy is the place to purchase top quality weaponry made from True-Silver

Eight Guilds exploit the activities of the Dungeon, it's a good idea to get into one of these and benefit from their services Every guild member gets a Spell Ring for casting spells, but it does have a limited life, but it can be recharged at a price! The list of spells is endless, some are listed in the manual

Encounters with other life forms can be a pleasure or an ordeal It depends sometimes on whether you surprise them or they surpise you! If a fight breaks out you have 4 (real time) seconds to choose an action Quick reflexes are essential Encounters don't necessarily mean fighting, you can talk to whom ever you meet offer them gifts in the hope of clues and rumours

The Dungeon is a game about exploring every crevice and cavern, going on quests of discovery but mostly about survival! If you've never tried a fantasy role playing game before, this could be the one for you

Druid

From Firebird
Disk £14 95, Cassette £9.95
For 800/800XL/130XE
Review by Bill Dyer

You take the part of the last Great Druid in a search for the four demon princes which have appeared through an inter dimensional gateway in the dungeons of Acamantor The evil princes take the form of skulls and are to be found lurking in the depths of the dungeon Only your strongest spell can destroy them and then only if you are touching the skull But the dungeons are dangerous to travel through, you will need all of your cunning and spells of invisibility and chaos to help you You must collect keys to enable you to progress from level to level, and you employ Water, Fire and Electrical spells to destroy the various hazards in your path Don't get wet yourself though, Druids are not all that keen on water! Dotted about the dungeon are chests containing new spells to aid you, in addition Pentagrams of life replenish your life essence Any baddie that attaches itself to you will drain your life essence

The best spell to find is the Golem spell This creates an ape like creature which becomes your servant and it provides protection for you by walking into your adversaries and destroying them (it also floats which is a distinct bonus for a Druid who hates water!) The Golem is controlled by pressing the A key (automatic mode) or a second player can control it via a joystick in port 2

Druid is a conversion from every other computer you can think of to the Atari, but don't let that fact detract you This conversion is one of the better ones. There are 8 levels to the dungeon, skulls only being found in the last four The top of the screen shows your status, i e how many fire, water or electric spells are remaining, how many key, chaos, golem spells left, current energy levels of yourself and the golem Your rating is also displayed, from halfwit, apprentice, acolyte through magic master and high priest to the ultimate Light Master

In general, Druid is a well constructed playable game with good graphics Maybe slightly overpriced in these days of budget titles but good value nevertheless!

Pirates of the Barbary Coast

From Cascade
Price £9 95 XL/XE, £12 95 ST
Review by Bill Dyer

The scenario is that your ship has been attacked by Bloodthroat the Pirate, who has taken your daughter Katherine hostage and is demanding 50 000 gold pieces in ransom You must therefore sail your ship around the barbary coast trading in silks cocoa, rum muskets, medicine, tea and wool buying and selling at various ports trying to make a profit. Or you can fight other ships and if you sink them you can plunder them for their booty Once you reach your target you can sail to the islands to bargain or fight Bloodthroat for the return of your daughter One of the islands is a treasure island where great wealth can be found, if you know where to look!

Whilst trading you must also keep your crew alive you do this by visiting the store in each port and replenishing with food You can also stock up on cannonballs, new crew members or buy information from the storekeeper If you are damaged at sea you can get your ship repaired, for a price, in any port

When at sea, if another ship is sited you can fight or flee, to fight though you have to go through a long sequence of loading your cannon in the correct order before they can be fired When the cannon are loaded they change to green, if they are knocked out by enemy fire they turn purple and cannot be used. Using a joystick controlled cursor, you can scroll along the 15 cannons trying to line up the enemy ship for a broadside You can raise or lower the elevation of your guns and if you score a hit, the enemy bursts into flames

In the final conflict with Bloodthroat you are confronted by his gigantic galleon at anchor in a lonely lagoon Your reactions must be swift in order to load your cannon and then you can give him a good broadside shot with all your cannon with luck you'll sink him and save Katherine.

SCRABBLE CROSSWORD

RUNS IN 48K
By Terry Wood - Reading

This board game is a cross between scrabble and a crossword On running the program the rules will be presented and you will be asked to select one of 5 games. Once selected many words are placed on the screen in a crossword type format The start of each word is printed in red whereas the rest of the word is in black Also on the screen are your first 7 letters to be used on the crossword, the letters allocated to the computer are also shown The computer decides who goes first. In the very centre of the crossword a flashing yellow cursor can be seen, you move this over the letter you wish to cover using a joystick in port 1 However you cannot cover just any letter, the first red letter of a word must be covered in order for other letters in the word to be covered and then they can only be covered in sequence Sounds complicated but once you start playing you'll get the hang of it You can lay two letters per go, and then the computer takes a turn, and two new letters are added to your set.

You score points by being the one to complete a word But watch out, you only get one point no matter how long the word, i e. a three letter word scores the same as a nine letter word. You finish the game when all letters have been covered

Type in the program, use the two check code letters at the start of each line with KEYO or TYPO otherwise ignore them Save off a copy to disk or cessette before running the program just in case you have made an error, we don't want all your typing wasted do we! Then run the program and enjoy the game.

Note· Anything underlined in this program should be typed in inverse.

```
GV 2 GRAPHICS 0
RG 3 DIM T$(10),POOL$(200),YILE$(1),PLAYER$(7),COMPUTER$(7),ENTRY(250,7)
MV 4 DIM MOVE(40,2),CHAR$(7)
GO 6 LINK=0:AF=1:NO=2:OF=3:OO=4:XP=5:YP=6:CTR=7
LC 7 ENT=0:SCO=1:TL=2
OI 8 GOSUB 900
UO 10 REM CONVERT CHARS TO MULTICOLOUR
HC 20 CH=PEEK(756)*256:C0=32*1024
RM 30 FOR C=33*8 TO 58*8 STEP 8
FS 40 FOR I=0 TO 7
MA 50 X=PEEK(CH+C+I)
VC 60 Y=170:X=170:A=85:B=85
AB 80 IF X>63 THEN X=X-64:Y=Y+16:A=A+32
RH 90 IF X>31 THEN X=X-32:Y=Y+4:A=A+8
KT 100 IF X>15 THEN X=X-16:Y=Y+1:A=A+2
JO 110 IF X>7 THEN X=X-8:X=X+64:B=B+128
EX 120 IF X>3 THEN X=X-4:X=X+16:B=B+32
OK 130 IF X>1 THEN X=X-2:X=X+4:B=B+8
BA 140 IF X=1 THEN X=X+1:B=B+2
RR 150 POKE C0+C+1,Y:POKE C0+C+I+512,Z
GN 160 POKE C0+C+I-256,A:POKE C0+C+I+256,B
AP 170 NEXT I:NEXT C
AA 180 FOR I=0 TO 7:POKE C0+I,85:POKE C0+I+256,85:POKE C0+I+512,170:POKE C0+I+768,170:NEXT I
YH 190 FOR I=0 TO 7:POKE C0+8*63+1,5:POKE C0+8*62+1,85*(I<3 OR I>4):NEXT I
EQ 200 REM SET UP DISPLAY LIST
PF 205 POKE 756,C0/256
GP 210 A=PEEK(560)+PEEK(561)*256
PH 220 POKE A,6:POKE A+1,PEEK(A+4):POKE A+2,PEEK(A+5)
BG 230 FOR I=3 TO 16:POKE A+I,5:NEXT I
AE 240 POKE A+17,65:POKE A+18,PEEK(A+30):POKE A+19,PEEK(A+31):POKE 752,1
TX 245 I=C0+60*8:POKE 1,170:POKE I+1,190:POKE I+2,190:POKE I+3,170:POKE I+4,85:POKE I+5,85:POKE I+6,85:POKE I+7,85
RD 246 POKE I+8,170:POKE I+9,190:POKE I+10,190:POKE I+11,170:POKE I+12,170:POKE I+13,190:POKE I+14,190:POKE I+15,170
RN 250 REM SET UP BOARD
BX 260 PRINT CHR$(125):SETCOLOR 0,9,10:SETCOLOR 1,13,10:SETCOLOR 2,8,8:SETCOLOR 3,3,4:SETCOLOR 4,12,4
TS 265 COLOR ASC("_"):FOR Y=0 TO 14:PLOT 38,Y:NEXT Y:POSITION 31,7:? "^^^^^^^^^"
UO 266 POSITION 32,6:? "YyQoUu":POSITION 33,8:? "MaEe"
NY 267 FOR I=0 TO 25:ENTRY(1,0)=0:NEXT I:USED=26
PB 270 POOL=0:I=0
WQ 275 RESTORE 300+INT(GAME)*50
PF 276 IF GAME=0 THEN RESTORE 550+INT(RND(0)*5)*50
TL 280 REMO X,Y,T$
CO 290 IF X<0 THEN 370
NB 300 C=128
DP 310 LAST=0
OY 320 NUMBER=ASC(T$(I,I))
NG 325 COLOR C+NUMBER-32:PLOT 2*X,Y
IN 330 COLOR C+NUMBER-64:PLOT 2*X+1,Y
NI 331 POOL=POOL+1:POOL$(POOL,POOL)=T$(I,I)
PV 332 NUMBER=NUMBER-ASC("A")
JP 333 ENTRY(USED,LINK)=ENTRY(NUMBER,LINK)
GN 334 ENTRY(NUMBER,LINK)=USED
WQ 335 ENTRY(USED,NO)=LAST
OU 336 ENTRY(USED,AF)=0
OK 337 ENTRY(USED,OF)=-1:ENTHY(USED,OO)=-1
YO 338 ENTRY(LAST,AF)=USED
GP 339 ENTRY(USED,XP)=X:ENTRY(USED,YP)=Y
BO 340 ENTRY(USED,CTR)=NUMBER
XJ 344 LAST=USED:USED=USED+1
CS 345 C=0:X=X+1
IR 350 IF LEN(T$)<2 THEN T$=T$(2):GOTO 320
PO 360 GOTO 280
XR 370 REM OOMM
XE 400 REM OOMM
TA 401 READ X,Y,T$
GU 410 IF X<0 THEN 1000
NK 412 C=128
OX 413 LAST=0
PN 415 NUMBER=ASC(T$(1,1))
UH 418 LOCATE 2*X,Y,X:IF C=0 AND X<>32 THEN 436
NG 420 COLOR C+NUMBER-32:PLOT 2*X,Y
IN 430 COLOR C+NUMBER-64:PLOT 2*X+1,Y
OI 436 NUMBER=NUMBER-ASC("A")
MV 439 SEARCH=ENTRY(NUMBER,LINK)
MH 442 IF GAMRCH=0 THEN SEARCH=USED:GOTO 450
NL 445 IF ENTRY(SEARCH,XP)=X AND ENTRY(SEARCH,YP)=Y THEN 470
AP 448 SEARCH=ENTRY(SEARCH,LINK):GOTO 442
OA 450 ENTRY(SEARCH,LINK)=ENTRY(NUMBER,LI
```

```
   NK)
SA 452 ENTRY(NUMBER,LINK)=USED
UA 454 ENTNY(SEARCH,AF)=-1:ENTNY(SEARCH,A
   B)=-1
FA 456 ENTNY(SEARCH,XP)=Z:EATNY(SEARCH,YP
   )=Y
JE 458 EATNY(SEARCH,CTH)=AUABEN
NN 460 POOL=POOL+1:POOL$(POOL,POOL)=T$(1,
   1)
OB 470 ENTNY(SEARCH,OF)=B:ENTNY(SEARCH,OB
   )=LAST
XH 472 EATRY(LAST,DF)=SEARCH
NV 474 LAST=SEANCH
QP 476 IF SEANCH=USED THEN USED=USED+1
WY 480 Y=Y+1:C=0
ON 490 IF LEA(T$)<>1 THEA T$=T$(2):SOTO 4
   15
WU 495 SOTO 480
OU 550 DATA 3,B,BANANA
EY 551 DATA 1,2,HAT,6,2,THAM
YA 552 DATA 11,3,PABE
KZ 553 DATA 6,4,CNOWN
NA 554 DATA B,5,AZLE
NK 555 DATA B,6,TNAILEW
Y8 556 DATA B,7,RADDISH
ZN 557 DATA 6,8,EAR
ZA 558 DATA 1,9,FISH
QB 559 DATA 9,10,VALLEY
NN 560 DATA 1,13,HELICOPTEN,12,13,LIP
ZX 561 DATA -1,-1,OONN
BQ 562 DATA B,4,HAIA
GI 563 DATA 2,7,OAISY,3,B,BOTTLE
YN 564 DATA 4,11,NN1T
ID 565 DATA 5,5,BUS
QL 566 DATA 6,B,AZTEC,6,7,HEDSEHOS
IY 567 DATA 9,6,TAA,8,11,CAT,9,2,NONEN
ZH 568 DATA 10,8,QUAKEN
AF 569 DATA 11,B,CHIP,11,5,PIN
LR 570 DATA 12,10,LILLY
IK 571 DATA 13,1,EAGINE
ZB 572 DATA 14,6,NALLY
IA 573 DATA -1,-1,EAO
GJ 600 DATA B,B,HARLEQUIN,11,B,TACK
BV 601 DATA B,2,NOSE,7,2,AUO,11,2,ALPS
ZA 602 DATA B,4,LOBE,5,4,LIGHTENING
ED 603 DATA 4,6,BIKE,9,6,BLOB
JI 604 DATA B,7,ZERO
AC 605 DATA 10,8,JEAEL
HB 606 DATA 3,9,STRAND
ZN 607 DATA 7,11,BAT,11,11,FACE
IK 608 DATA B,12,BLUE
NJ 609 DATA 3,13,EXPLOSION
BT 610 DATA 1,14,VAT,11,14,TAEE,-1,-1,-BO
   NA
QY 611 DATA B,B,HANDLE,B,7,ZIGZAG
QC 612 DATA 2,B,NASPBERRY
OA 613 DATA 3,9,STWEET
SH 614 DATA 5,B,QUILL,5,11,CAPE
BO 615 DATA 6,4,1AK,6,8,JAN
OA 616 DATA 7,11,BLOT
TA 617 DATA B,B,HOUGHT,B,7,LIO
UK 618 DATA 9,10,STAIN
VN 619 DATA 11,B,TNAIN,11,11,FONT
PJ 620 DATA 12,6,BOAL
SA 621 DATA 13,B,CAP,13,11,CONE
GF 622 DATA 14,4,GNU,14,8,LANE
HN 623 DATA -1,-1,EAO
NV 650 DATA 4,B,CHASN,11,B,FOAM
ZU 651 DATA B,2,BPAANEH,8,2,THHOME
YN 652 DATA 3,4,ALE,7,4,TNAIN
KC 653 DATA B,5,ZIP
OZ 654 DATA 10,6,CLOAK
TC 655 DATA B,7,CLOCK,6,7,CLOTH
DA 656 DATA 3,9,AACE,B,9,JASUAN
DB 657 DATA B,1B,BEEF
AV 658 DATA 4,11,TAIAE,10,11,BNESS
HT 659 DATA B,12,BIT
AI 660 DATA I,13,TONIC
NH 661 DATA 7,14,EQUATDA
AB 662 DATA -1,-1,DOWN
LV 663 DATA B,7,CWAB
AJ 664 DATA 1,1,SPAAIEL,1,1B,EXIT
BN 665 DATA 2,12,TON
TT 666 DATA 3,6,SCAMF
BC 667 DATA 4,B,CANDLE,4,11,TRIP
JZ 668 DATA 5,9,COW
YQ 669 DATA 6,8,AIN
QS 670 DATA 7,4,TABLE,7,10,GWAME
DO 671 DATA B,8,NOTON
HO 672 DATA 9,7,TAAP
BV 673 DATA 10,4,INCH
TT 674 DATA 11,B,FNONN,11,11,NOOT
IF 675 DATA I2,B,VALE
VP 676 DATA 13,B,AMENICAN,13,11,BPUN
IN 677 DATA -1,-1,END
BU 700 DATA 1,B,ONEATEA,9,8,CDCDA
QZ 701 DATA B,2,HAIZE,6,2,LEAVES
XA 702 DATA 11,3,TWO
NO 703 DATA 4,4,BALE
UB 704 DATA B,5,KEY,7,5,CLAYMONE
NP 705 DATA B,7,THREBHEN,12,7,PES
ED 706 DATA 7,9,CLOTH
NL 707 DATA B,1B,UAPINE,12,1B,OOT
KK 708 DATA 7,11,TILE
FA 709 DATA 2,12,CANANY,10,12,ALIKE
OB 710 DATA 5,14,FEET,10,14,LADY
ZP 711 DATA -1,-1,OOAN
PG 712 DATA B,2,NANKET,8,9,QUANTZ
VN 713 DATA 2,B,WHIP,2,5,YARD,2,10,PACK
FA 714 DATA 4,2,ENGLISH,4,10,NUNB
AO 715 DATA 6,B,EEL,6,12,RYE
NN 716 DATA 7,2,ELECTNICITY
ZF 717 DATA 9,B,CAVE,9,5,AZOLOTL
LK 718 DATA 11,8,COSTUME
CI 719 DATA I3,8,ARAGUN,13,9,JOCKEY
BN 720 DATA 14,5,EOB
HN 721 DATA -1,-1,END
PL 750 DATA 7,B,JAPAA
KO 751 DATA 1,1,SILVER
YK 752 DATA 9,2,LAMP
BZ 753 DATA B,3,BUAY,5,3,ANKLE
NV 754 DATA 10,4,FEVER
FH 755 DATA 8,5,TEETH,6,5,ANTLER
BP 756 DATA 4,7,POOL,9,7,STNAP
DB 757 DATA B,8,TNAN
LO 758 DATA 5,9,TAX
QZ 759 DATA B,10,ULCER,10,10,HAZDA
FB 760 DATA 6,12,1ADIA,12,12,CAT
FP 761 DATA B,13,QTADIUN
NO 762 DATA 6,14,PNOJECTDA
AF 763 DATA -1,-1,OOAN
KL 764 DATA B,B,TOURIST
JD 765 DATA 1,1,SOUVENIN
NN 766 DATA 2,B,AACH
VV 767 DATA 3,B,CLAY
UF 768 DATA 4,5,HIP,4,10,HADIO
AK 769 DATA 5,B,PETAL
QC 770 DATA 6,5,ANOOAK,6,12,IMP
OL 771 DATA 7,2,SKIA
NE 772 DATA B,10,HOOEO
AE 773 DATA 9,B,POLE
ZV 774 DATA 10,4,FEATHEN,1B,12,AGE
TP 775 DATA 11,B,NUNBEN
ZL 776 DATA 12,12,COT
AH 777 DATA 13,4,EGYPT
AH 778 DATA 14,B,AATEN,14,9,NHITER
IU 779 DATA -1,-1,END
YT 80B ? CHN$(125);"          CROSS WORDS
   "
CN 81B ?
QL 82B ? "Computer choonan who wIll start
   ."
CR 83B ?
JZ 84B ? "Eech player can place two tilan
   ."
BO 85B ? "To place your tilan, nova the"
UV 86B ? "flashing yellow tile with the"
YJ 87B ? "joystick and preee the trigger.
   "
DB 88B ?
SV 89B ? "If you or the computer can't go
   , then"
KC 90B ? "a beep is heard and the other"
ED 91B ? "player continues."
CQ 92B ?
HN 93B ? "If you can't go any tilns, then
   "
HT 94B ? "all the tilan are put back and
   eeven"
JL 95B ? "more teken."
CY 96B ?
NS 97B ? "Select game 1 - 5, 8 for comput
   er"
NH 98B ? "choice ";:TAAP 98B:INPUT GAME
ZQ 985 IF GAME<B OR GAME>5 THEN 98B
BQ 99B ? CHN$(125);"          PLEASE NA
   IT"
KK 992 THAP 5BBB
AM 995 NETUNN
BT 1BBB NEN PICK TILEG
```

```
OY 1005 ONNX=I-I
ZM 1010 FOR I=1 TO 7
VD 1020 GOSUB 4000:PLNYER$(I)=TILE$
VD 1020 GOSUB 4000:PLRYEN$(I)=TILE$
LG 1030 GOSUB 4000:LET COMPUTER$(I)=TILE$
FB 1040 NEXT I
RF 1045 CSCORE=0:PSCONE=0:GOSUB 9500
RO 1047 L=I200:IF RRO(0)>0.5 THEN L=I050
FR 1048 GOTO L
EK 1049 GOSUB 9500
ZG 1050 GOSUB 6000
RZ 1200 CHRR$=PLAYER$
CO 1210 FOR TRY=1 TO 2
XR 1220 GOOUB 3000
YR 1230 IF OOPT<0 THEN I450
AR 1240 GOSUB 9000
DX 1250 FON I=0 TO OOPT
ZK 1260 SERRCH=MOVE(I,ENT)
DA 1270 IF ENTRY(SEARCH,XP)=X RNO ENTRY(S
   EARCH,YP)=Y TNEN I310
FR 1280 NEXT I
BO 1290 FOR F=40 TO 00 STEP 5:SOUNO 0,F,1
   2,0:REXT F:GOUNO 0,0,0,0
PS 1300 SOTO 1240
LO 1310 TINO=1:GOSUB 7000
CP 1320 PLRYER$(K,K)=" "
NA 1330 IF TRY=1 THEN GOSUB 9500
XB 1340 GOSUB 4000:PLNYEN$(K,K)=TILE$
YA 1350 SC=POCORE
WI 1360 Y0=3
FP 1370 SOSUB 9900
XH 1380 PGCORE=PGCORE+SCORE
SC 1390 IF PLRYER$="       " AND COMPUTER
   $="       " THEN 1600
RX 1400 NEXT TRY
QO 1405 GOSUB 9500
PN 1410 GOTO 1050
UR 1450 FOR F=I0 TO 0 OTEP -1:GOUNO 0,20,
   12,F:REXT F
DO 1460 GOSUB 9500
VG 1490 IF TRY=2 THEN 1050
T0 1500 GOSUB 9700:PLAYER$=CHRR$:GOSUB 95
   00
PO 1510 GOTO 1050
RJ 1600 FOR R=1 TO 500:REXT R
EN 1610 ? $6;CHR$(125);
BE 1620 IF PSCONE<CSCORE THEN POSITION 13
   ,7:? 06;"NwEe  OdRrAaNw";
UR 1630 IF PSCORE>CSCONE THEN POSITION 13
   ,7:? $6;"YyDoUu  NwIiNn";
IO 1640 IF PSCORE<CSCORE THEN POSITION 15
   ,7:? $6;"II  RwIiNn";
RY 1650 FOR R=1 TO 500:REXT W
RB 1660 GOTO 240
OB 3000 RER FIRO OPTIORO FOR TILE8
KX 3100 I=-1
AN 3110 FOR J=1 TO 7
OV 3120 NURBER=RSC(CHRR$(J,J))-ABC("R")
NO 3130 IF NUMBER<0 THEN 3800
VI 3140 SEARCH=ENTRY(NURBER,LIRK)
OJ 3150 IF BEARCH=0 THEN 3800
HB 3160 ALOC=0:SCONE=0
FR 3170 IF ENTRY(SEARCH,R0)<>0 THER 3500
UN 3180 I=I+I:NLOC=1
PY 3190 MOVE(I,ENT)=SEARCH:MOVE(I,TL)=J
IN 3200 LET NEXT=ENTRY(SEARCH,RF)
PQ 3210 IF NEXT=0 THER SCORE=2
RH 3220 IF ENTNY(NEXT,AF)<>0 THEN 3500
AG 3230 FOR K=I TO 7
PI 3240 IF K=J OR CHAR$(K,K)=" " TNER 327
   0
JP 3250 IF ENTNY(NEXT,CTR)=RSC(CHRR$(K,K)
   )-ASC("R") TNER SCORE=SCORE+I:SOTO 350
   0
GR 3270 NEXT K
KK 3500 IF ENTRY(SEARCH,O8)<>0 THEN 3700
RQ 3510 IF ALOC=0 THEN I=I+1:ROVE(I,ENT)=
   SEANCH:MOVE(I,TL)=J:ALOC=I
LV 3530 LET REXT=ENTRY(GERRCH,OF)
KR 3540 IF REXT=0 THEN SCONE=SCORE+2
FB 3550 IF ENTRY(REXT,DF)<>0 THEN 3700
8H 3560 FOR K=I TO 7
XT 3570 IF K=J OR CHRR$(K,K)=" " THER 359
   0
PY 3580 IF ENTRY(NEXT,CTR)=AOC(CHRR$(K,K)
   )-R8C("A") THER SCORE=SCORE+I:GOTO 370
   0
SY 3590 NEXT K
NQ 3700 IF ALOC=1 THER ROVE(I,SCO)=SCORE
FQ 3710 SERNCH=ENTNY(SEANCH,LINK)
AF 3720 GOTO 3150
FS 3800 NEXT J
PJ 3900 OOPT=I
SU 3990 RETURN
NZ 4000 REN PICK TILE FNON POOL
LP 4010 IF POOL=0 THEN TILE$=" ":RETURN
XX 4020 T=IRT(RN0(0)*POOL)+I
EG 4030 TILE$=POOL$(T,T)
FR 4040 IF T<>POOL THEN POOL$(T)=POOL$(T+
   1)
IA 4050 POOL=POOL-I
AU 4060 RETUNR
VP 5000 GRAPHICS 0:PRINT "ERROR ";PEEK(I9
   5);" AT LINE ";PEEK(186)+PEEK(187)*256
EU 5010 ERO
NB 6000 RER COMPUTER PLAY
CN 6010 CHAN$=COMPUTER$
CH 6020 FOR TRY=I TO 2
XV 6030 GOSUB 3000
ZF 6040 IF OOPT<0 THEN 6230
OA 6050 RIN=-1
EG 6060 FOR I=0 TO OOPT
EO 6090 IF ROVE(I,OCO)>RIN THEN RIN=MOVE(
   I,0CO):HIGH=I
FV 6090 NEXT I
NK 6100 I=HIGH
HV 6120 TINO=10:GOSUB 7000
UU 6130 LET COMPUTER$(K,K)=" "
RE 6140 IF TRY=1 THEN GOSUB 9500
PI 6150 GOSUB 4000:LET COMPUTEN$(K,K)=TIL
   E$
TR 6160 SC=CSCONE
OS 6170 Y8=10
FT 6180 GOSUB 9900
NY 6190 CSCONE=CSCORE+SCORE
RE 6200 IF PLAYER$="       " AND COMPUTER
   $="       " THEN 1600
OB 6210 NEXT TRY
DU 6215 GOOUB 9500
RP 6220 GOTO 6410
UR 6230 FOR F=10 TO 0 GTEP -I:GOURO 0,20,
   12,F:NEXT F
OJ 6240 GOSUB 9500
YZ 6290 IF TRY=2 THEN 6410
RU 6300 GOSUB 9700:LET COMPUTER$=CHAR$:GO
   SUB 9500
RP 6410 RETURN
NP 7000 REN PLRY A LETTEN
IB 7005 SCORE=0
IT 7010 K=NOVE(I,TL)
ON 7020 TILE$=CHNR$(K,K)
CH 7030 CHNN$(K,K)=" "
HX 7040 NOVE=ROVE(I,ENT):X=2*ENTRY(NOVE,X
   P):Y=ENTRY(NOVE,YP)
RL 7050 FOR L=I TO TINO
8G 7060 POSITION X,Y:? #6;CNR$(ASC(TILE$)
   +96);CHR$(NSC(TILE$)+64);
AS 7070 FOR G=8 TO 0 STEP -1
EF 7080 SOURO 0,20,10,G
KC 7090 NEXT S
VO 7100 POSITION X,Y:? #6;TILE$;CHR$(ASC(
   TILE$)+32);
0J 7110 FOR N=0 TO 8:NEXT R
GK 7120 NEXT L
IQ 7200 IF ENTRY(NOVE,AF)=-1 TNEN 7300
ST 7210 PRIOR=ENTRY(MOVE,A8):LET NEXT=ENT
   NY(NOVE,NF)
8N 7220 IF PNION=0 NNO NEXT=0 THEN 7250
JO 7230 IF PRIOR<>0 THEN ENTRY(PRIOR,NF)=
   NEXT
YR 7235 IF REXT<>0 THEN ENTRY(NEXT,A0)=PR
   IOR
RI 7240 GOTO 7300
FK 7250 SOURO 0,40,10,8:FON N=1 TO 50:NEX
   T N
EZ 7260 GOUNO 0,50,I0,8:FOR N=0 TO 50:REX
   T R
OL 7270 SOURO 0,0,0,0:SCORE=SCORE+1
D8 7280 ERTRY(MOVE,AF)=-I:ENTRY(MOVE,A8)=
   -1
NH 7300 IF ENTRY(MOVE,OF)=-I THEN 7400
RX 7310 PRIOR=ENTRY(ROVE,08):LET REXT=ENT
   NY(MOVE,OF)
OS 7320 IF PRIOR=0 ARO NEXT=0 THER 7330
NR 7330 IF PRIOR<>0 THEN ENTRY(PRIOR,OF)=
   NEXT
CO 7335 IF REXT<>0 THER ENTRY(REXT,08)=PR
   IOR
RN 7340 GOTO 7400
FN 7350 SOUNO 0,40,I0,8:FOR R=1 TO 50:NEX
```

```
   T R
FQ 7360 SOUND 0,50,10,8:FOR N=0 TO 50:REX
   T N
DN 7370 SOUND 0,0,0,0:SCORE=SCORE+1
UR 7400 SEARCH=ENTRY(ROVE,CTR)
ZZ 7410 IF ERTRY(SEARCH,LINK)=ROVE THER E
   NTRY(SEARCH,LINK)=ERTRY(ENTRY(SEARCH,L
   INK),LINK):GOTO 7500
GP 7420 SEARCH=ERTRY(SERRCH,LINK):GOTO 74
   10
TK 7500 GOTO 7900
RV 7900 RETURR
UR 8000 RER FIND HIGNEST TILE TO PLRY
RU 8010 RRX=-1:I=-1
XY 8020 IF OOPT<0 THEN RETURR
EO 8030 FOR K=0 TO OOPT
VD 8040 IF MOVE(K,3)>MAX THER RRX=ROVE(K,
   3):I=K
GR 8050 REXT K
RY 8060 RETURR
CK 9000 X=14:Y=7
GF 9010 LOCATE X,Y,I:LOCRTE X+1,Y,J
NU 9020 IF I-128*(I>=128)<64 THER POSITIO
   N X,Y:? 06;CHR$(I+32+64*(I=32));CHR$(J
   +96);:FOR N=1 TO 5:REXT R
PK 9025 IF I-128*(I>=128)>=64 THEN POSITI
   OR X,Y:? 06;CHR$(96);CHR$(96);:FOR N=1
    TO 5:REXT R
IN 9030 POSITIOR X,Y:? 06;CHR$(I);CHR$(J)
   ;:FOR N=1 TO 5:REXT N
AZ 9035 IF PEEK(53279)<>7 THEN GOSUB 9200
   :GOTO 9035
BC 9040 Z=STICK(0):IF Z<>15 THER POKE 77,
   0
HE 9050 IF Z=14 OR Z=10 OR Z=6 THEN Y=Y-1
   :IF Y<0 THEN Y=14
TV 9060 IF Z=9 OR Z=13 OR Z=5 THEN Y=Y+1:
   IF Y>14 THER Y=0
DH 9070 IF Z=9 OR Z=11 OR Z=10 THER X=X-2
   :IF X<0 THER X=28
JR 9080 IF Z=6 OR Z=7 OR Z=5 THEN X=X+2:I
   F X>28 THER X=0
JF 9090 IF STRIG(0)=1 THER 9010
DH 9095 POKE 77,0
RZ 9100 X=X/2:RETURN
KE 9200 FOR Z=0 TO OOPT
LD 9205 SERRCH=ROVE(Z,ENT)
BN 9210 XR=ENTRY(SEARCH,XP)*2:YR=ENTRY(SE
   RRCH,YP)
EB 9220 LOCRTE XR,YR,I:LOCRTE XR+1,YR,J
FI 9230 POSITION XR,YR:? 06;CHR$(96);CHR$
   (96);:FOR R=1 TO 5:REXT R
HB 9240 POSITION XR,YR:? 06;CHR$(I);CHR$(
   J);:FOR R=1 TO 5:NEXT N
MV 9250 NEXT Z
AN 9300 RETURN
RB 9500 FOR I=1 TO 7
RB 9510 Z=RSC(PLRYER$(I,I))+64*(PLRYER$(I
   ,I)=" "):J=I*2:X=J+29-(I>4)*7:Y=(I>4)
UO 9520 COLOR 2:PLOT X,Y:COLOR 2+32:PLOT
   X+1,Y
LU 9530 Z=RSC(COMPUTER$(I,I))+64*(COMPUTE
   R$(I,I)=" "):Y=13+(I>4)
UU 9540 COLOR 2:PLOT X,Y:COLOR 2+32:PLOT
   X+1,Y
FR 9550 REXT I
BJ 9560 RETURR
NS 9700 RER RETURR TILES TO POOL
RI 9710 FOR I=1 TO 7
OD 9720 IF CHAR$(I,I)=" " THER 9750
HJ 9730 POOL=POOL+1
XJ 9740 POOL$(POOL)=CHRR$(I,I)
GA 9750 NEXT I
RX 9760 FOR I=1 TO 7
BA 9770 FOR I=1 TO 7
GS 9780 GOSUB 4000:CHAR$(I,I)=TILE$
GN 9790 NEXT I
RX 9800 RETURR
DF 9900 RER DISPLRY SCORE
RO 9910 IF SCORE=0 THER RETURR
JL 9920 FOR OFFSET=SC TO SC+SCORE-1
NC 9930 Y=INT(OFFSET/16)
SQ 9940 X=OFFSET-INT(OFFSET/8)*8
ZN 9950 CHIRK=IRT(OFFSET/8)-Y*2
SI 9960 POSITIOR X+3I,Y+YD:? 06;CHR$(220+
   CHIRK)
EL 9970 REXT OFFSET
BX 9980 RETURR
```


NEXUS

ARE LOOKING FOR NEW BLOOD!

NEXUS PRODUCTIONS
are looking for experienced programmers and graphic artists to work on a range of innovative 16 Bit games

OR CALL IAN ELLERY ON: 01-658 5723 For more details

NEXUS PRODUCTIONS LTD
DSB House 30 The High Street Beckenham Kent BR3 1AY

ST LIBRARY

Librarian: Mike Stringer

Introduction

Allow me to tell you how the ST Library is going to be structured. Listed here are the disks currently available.

The disks that I will be sending out are DS/DD but will be formatted for single sided use. Where the program requires 1 Meg formatting, these disks will be clearly marked and no additional fee will be requested. In other words the fee will be the same irrespective of the size of the program(s).

In some instances the files may be compressed. The necessary Archiving program will always be included on the disk, including the necessary info to allow you to convert them back to normal. In this way I will be able to put up to the equivalent of 500K of files on one half meg disk.

In addition to the files I will also include, if space permits an up to date list of the library. The reason behind this is to keep you up to date at all times, you will not have to wait the three months or so for Monitor to arrive.

Each disk will be filed under a heading according to the subject which the program/files relate. For example LP1 is a Language disk, the subject is Pascal and it is the first in this particular section. Or MMS1 is a MIDI disk containing files for Music Studio again number 1.

There will also be a Support section which is intended to be used with programs/files for use with existing Commercial Software. For example, templates for VIP. Fonts for word processors or Printer Configurations and so on.

MIDI support files will be contained within the MIDI section because of the nature of the subject. I have given one example, but others already include Casio CZ Voices, 36 banks of voices for the Yamaha DX7 with the DXDROID etc.

As other sections become available they will be introduced. Wherever possible, programs and files will be segregated to maintain integrity. If there is a demand for a mixture I will try to oblige, this will be the exception, not the rule.

What to do

The club has laid out a great deal of money to get the Library off the ground and in order to recoup these costs and to obtain new material it is necessary to make a small charge. There are two services currently available. The first you provide the disk with your request and the fee is £3.50. The second, we provide the disk (DS/DD) when the fee is £5.50. This includes all necessary return postage and packing.

Any member who submits material will have his disk returned, the contents having been copied into the Library, to be replaced by something very useful (or a request of your own) as a form of thanks. Please remember that if you do submit any material it must qualify for the description of Public Domain, as something similar, i.e. no ripped off Commercial Software will be tolerated.

If at any time you wish to obtain the latest complete library list, just send a disk and £1.00, or just send £2.50 and we will supply a disk with the list recorded onto it.

The ST Library is for subscribers only.

CONTRIBUTIONS

I would first of all like to thank those members who have very kindly sent new material to the Library. This material is currently being classified into their respective sections and where single items have been submitted they are awaiting similar material to fill up a disk. Some of the material has been written by the members and is very good. I will make suitable references to these programs when they are introduced.

There are no fewer than thirty one additional disks to be included in this issue. They come from the States, Canada and Australia in the main and reflect my intentions to make available good quality and new programs for your edification.

Some of these are worthy of note. You will notice a considerable number of Language disks. There are some more Utility disks covering numerous subjects. Of note to the MIDI enthusiasts, especially although not specifically CASIO owners, the MSAS series of four disks should be of interest to you. They have been created by young John Bond who has combined Music Studio files and pictures to create an interesting product. Incidentally you do not need to use Music Studio to appreciate them! They are stand alone files.

WITHDRAWALS

I would like to offer equal thanks to those members who have shown so much interest in the material being offered. I hope that your choices have given you much enjoyment!!

PCW SHOW

It was a pleasure to meet so many of you at the Show. I am very sorry that you did not have the opportunity to see and try out the items being offered in the library. There were two reasons for this. Although ATARI very kindly gave us space, it was a very small allocation. No fewer than twelve user groups were invited to share six tables and six chairs! To say that it was cramped is putting it mildly! Secondly, there was a directive from ATARI that absolutely no copying of any kind was to take place on the stand! To have been only slightly efficient, we would have liked to use two ST's plus a small stock of our magazines and other literature. Can you imagine what this would have looked like squeezed into the allotted space? We will try very hard to provide a much better service at the next show in November.

I had a couple of requests to give a detailed listing of each disk. I am afraid that this is a request that is impossible because of space. I would recommend that you send in for the disk based listing. This does list all files but I would strongly recommend that you do not print it out but rather load the file into a suitable Word Processor where you can examine it at your leisure, using SEARCH commands to find specific files and programs.

Here follows a short description of the programs received since the last issue:

AGR 1

51 MONO files and programs including GRAPH FRACTAL SPHERE MANDEL. 1/2 MB

ADEMO 2

A fabulous graphics demo with two stainless steel balls scrolling chess board and rotating spheres. 1/2 MB colour

ADEMO 1

Colourful continuous graphic/sound demo of ATARI software and features. 1/2 MB colour

ARTMETA 1

223 META files (.PIC) 1/2 MB.

ARTMETA 2

A further 257 (.PIC) files 1/2 MB

COM 1

28 communication files including STERM, KERMIT, STXMODEM, STTALK. 1/2 MB.

LFTH 1

44 FORTH (.FTH) files including FORTH TOS KERNEL, META, LINEA. 1/2 MB

LXLSP 1

XLISP language disk containing 4 items. 1/2 MB

LLOGO 1

44 (.LOG) files including SQUIRM, DEMOLOG, FLAG, DOODLE. 1/2 MB

LMOD 1

40 MOD, DEF, LNK files including ARCSHELL, LIEA, FORMAT. 1/2 MB

LMOD 2

More .DEF .MOD files including ARGHANDLE, ARGTEST, FILEIO. 1/2 MB

LPAS 1

35 .PRG .PAS files including CHECKERS, CUBE, PRINT. 1/2 MB

LPAS 2

54 .PAS .PRG .TOS files including PFORMA, SHELL, MORGAGE. 1/2 MB.

LPAS 3

55 files including OSS PASCAL, SMALL TUTORIAL, ATARTREK. 1/2 MB

LSTB 1

83 (.BAS) files including SKYSCAPE, DISPLAY. 1/2 MB

MDXORG 1

Run time demo from Lawrence Wilkes.

DXORGANISER is a voice librarian for the Yamaha DX7 including new banks (1500 voices). 1/2 MB

MVAR 1

20 items including CZVOICE, CASIOL, MIDITO, SOUNDKEY. 1/2 MB

MVAR 2

39 items including DLXPIANO, CASLIB, MIAMI. 1/2 MB

MSAS 1

10 John Bond Music Studio/Picture files including FLASHDANCE, TAKE FIVE, GLADIATOR. Does not require Music Studio. 1/2 MB colour

MSAS 2

10 similar files as above including CLASSICAL GAS, TOCCATA, CANON. 1/2 MB colour

MSAS 3

9 similar files as above including MUSIC BOX, DANCER, SEMPER, MASH. 1/2 MB colour

MSAS 4

10 additional files to above including LONELY, ALLA TURKA. 1/2 MB colour

PRINT 1

77 fonts including FONTRIX and FONTLOAD accessories. 1/2 MB

UTIL 5

50 items including SECED, STSQ, DEGZNEO. 1/2 MB.

UTIL 6

35 items including ADDRESS, SOUND, DSPEED, TPRINT. 1/2 MB

UTIL 7

44 items including DISKMON, ENCRYPT, PRINTDIR. 1/2 MB

UTIL 8

45 items including CZVOICE, CASIO70, FORM, GRAFCON, EZSQUEEZE. 1/2 MB

UTIL 9

52 items including MCALC, SOUND, KDC, KDCV2. 1/2 MB

VIPTEM 1

19 VIP templates including LOAN, REPORT, TAX. 1/2 MB

VIPTEM 2

37 additional templates including CPATH, INVENTORY, PAYROLL. 1/2 MB

VIPTEM 3

31 additional templates including CASHPROJ, BSINC, AMORT. 1/2 MB.

Requests should be sent to the address above.

WS REVIEWS ST REVIEWS ST REVIEWS ST

mining installations but .. oh dear! The Terrorpods hava just destroyed tham all! Your puny Laser systems only hava the effect of temporarily stopping the Terrorpods They fold themselves up into a little box that hes only Doctor Who's Tardis as an equal in storing a lot of matter in a little space and take off for the Mother Ship only to return a short while later No..what you need to destroy a Terrorpod is a missile armed with Detonite Of course you don't have an endless supply of Detonite either but you

can also trade with a whoops there goes another mining installation!

While all this is going on the Mother Ship is intent on sending you off to the great Space Invader machine in the sky by bombarding you with deadly energy-eating missile thingies Each time you are hit by one of these your anergy goes down faster than the value of shares in Westland If you are hit four times then the game is over One way out of this mayhem is to warp off to another colony where the Terrorpod advance hasn't yet started in earnest This is of course only if you have enough energy to make the jump Just about everything you do depletes your supply of something or other and it is a constant battle just to get enough supplies to carry on with the game Because of this a very large part of the game is the trading aspect. Each mining station places a different value on the various commodities available and if you are clever enough and have a good enough memory you can make a handsome 'profit' by judicious buying and selling In practice I found it difficult enough just staying alive in a 'war' zona without worrying about whether I was getting a good deal or not!

This game is excellent and could well become the 'Elite' of the Atari ST It is vast in concept and awesome in strategy There are so many things going on at one time it is difficult to know what to do If you prefer a mindless blast em to bits type game then this one is not for you There is certainly a shoot em up aspect to it but if you just proceed along this path then you'll last about as long a Sinclair C5 in a six foot snowdrift At some point you are forced to attempt to replenish your supplies and then the fun starts! The Terrorpods are merciless and do not let up in their destruction of the colonies I quickly found things getting out of control as more and more Terrorpods landed and more and more of my vital supply depots were decimated You have the ability to quickly rebuild dumps by using some Quaza But inevitably you run out of this as well

The graphics are suparb, from the highly colourful and detailed background, to the animated and devilish Terrorpods themselves My favourite bit is when you have to launch a Detonite missile You first arm it by pressing 'A' (so long as you have enough Detonite) and then you press 'E' to call up the missile system This appears as a drop-down window In fact a lot of the displays drop down such as tha Radio interface, the Trading screen and the Status displays You use the joystick to sight and lock onto your Terrorpod target. This actually appears as a fixed window scrolling over a portion of the main screen display which is still thera in the background and believe me the effect is quite breathtaking When you have locked on you press the fire button and the missile fires You then have a countdown to impact as the missile travels its course during which time it is apt to swerve from its target You have to keep the horizontal and vertical sights fixed within the crosshairs a difficult job All being well you experience a blinding explosion and tha Terrorpod is satisfyingly blown to bits!

The game goes out of its way to be helpful to you, If you press the 'Help' key at any time then the game is paused and you go to a game management screen. From here you can set the difficulty level, restart the game or even Load and Save a game. I thought that this was a particularly useful feature since even a mere beginner such as myself managed to keep going for over half an hour and I can foresee that an experienced player could go for hours! Another nice feature is that you can call up a display of all the key functions, so if you suddenly forget how to commence trading or fire that Detonite missile at a crucial time, you can easily find out how

One of the things that really impresses me about any game is if it saves the High Score table to disk And yes, Terrorpods does this (although the number of times I actually get on it is severely limited!) I found it very difficult to master this game. I suspect that it would take a very, very long time to become proficient at it which of course makes it excellent value for money I don't think the trading aspect of the game was explained well enough in the accompanying documentation but perhaps this was deliberate as it became fairly obvious with continued playing as I

learnt more and more about what I was supposed to be doing. There is no doubt in my mind that I got actually nowhere I didn't manage to get hold of one Terrorpod part and my trading abilities were non exstent But I can see that this is an addictive game, you just have a lot of learning to do Tantalisingly mentioned in the Press Release that accompanied the review copy was the fact that there is some kind of 'end game' What form this takes I have no idea but I suppose it could be a 'Clash of the Titans' fight with the Mothership or something equally awe inspiring

All in all, I think that this is an extremely good game I've never been much good at games where you have to do a bit of thinking but I returned to this one over and above the call of a reviewers duty! The packaging is excellent and the scope vast. As they say in the movies this one will run and run!

GFA-Draft

From Glentop
Price £99 95
Review by Athena

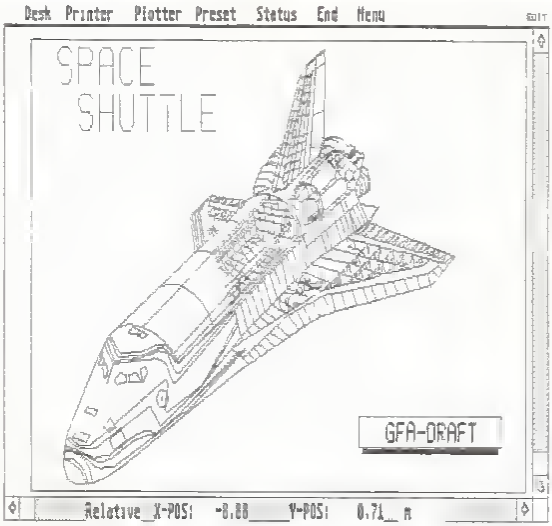
Computer Aided Design (CAD) applications used to be the sole domain of big expensive computers with masses of cables linked to huge plotters all trundling slowly up and down pieces of paper With the release of the ST there has been a marked increase in the numbers of programs which can simulate comprehensive CAD packages Many of these fall into the 'paint' type programs whilst some are obviously aimed at the professional with a matching price tag

GFA DRAFT is a powerful GEM based drafting program, quite similar to various drawing packages available for the ST, but with added features such as automatic scaling, user definable symbols the rotation and skewing of drawings and symbols Up to 255 drawing levels are available with ten levels displayed at any one time To bring up the rear, comprehensive printer and plotter routines have been included to add that finishing touch to your drawings

Two menu bars are used to select the numerous commands available from the package The first bar holds six menus comprising of file storage and retrieval, types and thickness of lines window functions, storage and retrieval of symbols, options and finally picture functions The second menu has printer routines, plotter routines, preset, status and end.

The best way in which to review GFA DRAFT is to take you through the menus one by one First things first, the manual pages are an awful red colour which makes it hard on your eyes and with programs like this where you need to spend a large amount of time reading and re reading sections it can become frustrating.

The drawing area is 658mm x 658mm which can be scaled from full size (1 1) down to miniature (1 1000), the default page size is 153mm wide by 88mm high Being GEM based, the page



Space Shuttle

can be viewed by scrolling the screen as normal GEM conventions The first real menu is the files, only four options here, read picture, store picture delete and rename the last two can come in handy as there s nothing worse than having to exit to GEM every time you wish to change a name or delete a file

Following on from files the next menu is lines, here there are four types of lines to choose from solid short dashes, long dashes and dotted All the line widths can be altered from 0 3mm to 4.5mm in 0 3mm graduations Lines are normally displayed all the same width regardless of weighting, however if you wish lines can be displayed with their actual thickness shown Lines which are meant to be next to each other can be 'locked' together sort of a 'snap to guides' This couples up with a squaring command that ignores small errors in placing the lines both vertically and horizontally For lines that have crossed, 'break line' will separate them into four individual lengths If you're like me when it comes to placing lines then you'll need to use the trim lines' command to get your edges neat and cut off the overhang

The window menu allows you to be selective when modifying your picture drawing a 'window' is performed by dragging a box around your chosen object, once framed the object is regarded as a picture' to the computer From this window, copying or moving are as simple as dragging the object to another location, however this is not just the only command available, objects can deleted rotated in 90 degree presets or in 1 degree increments reflected on either the X axis or Y axis, stretched and compressed Skewing (leaning to the left or right) is a doddle and by a little intuition, objects can be skewed vertically by first rotating the window 90 degrees, skewing the object then rotating the

window back to its original position.

Still within the window menu the contents can be zoomed by a factor between 0 01 and 9 99 A factor of 0 1 reduces the object to a tenth of its original size where a factor of 2 0 will increase the object to twice its original dimensions Whan it comes to deleting objects from a window, another parameter comes into force, either lines and/or text can be deleted

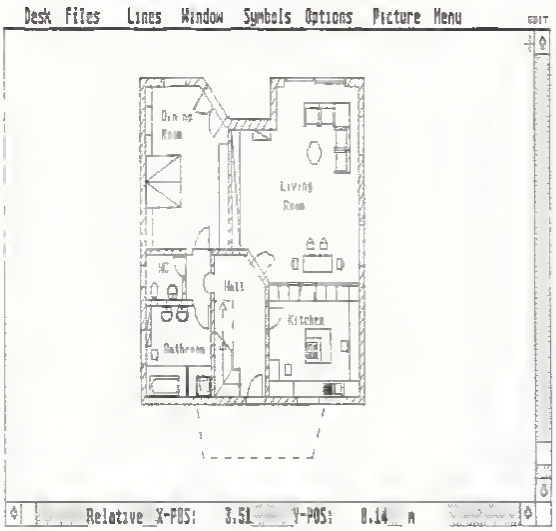
The fourth menu looks after the symbols these are a collection of lines and/or text that has been defined as a self contained logical unit. Any object can be defined as a symbol, just drag a window round it and save it under the save symbol option, each symbol has an eight letter filename with SYM as the extender This menu has a similar style to the files menu load, save, rename and delete all these act as youd expect them to. Over 100 symbols are already stored on the accessory disk ranging from electronics to flowcharts The good thing about symbols is that you can use them over and over again, if you find that a particular one is used frequently you can assign any of the function keys to load it up and display it immediately on the screen Altogether ten symbols can be defined with the function keys, unfortunately they cannot be stored to disk

The penultimate menu houses the standard drawing commands such as rectangle, circle and ellipse There are more complex functions encompassed within the menu though Segments of a circle or ellipse can be easily constructed and a very easy way of drawing curves is to use the arc 3 points command Just click on any three points on the screan and the computer draws the curve instantly Curves can be any size from horseshoe to serpentine

Creating rectangles are usually quite easy but when you want something different the round corners option can be extremely visual Corners can be rounded from 1 to 99mm radius and to make things even more impressive the lines representing the corners don't have to meet, GFA-DRAFT automatically adjusts the lines to suit

There can be times whare lines need to be perfectly perpandicular (right angles) to an existing line and by using this option you can pull a perpendicular to either side of the line This may sound complicated but it's really easy to operate The function angle to a line works in the same way as for drawing perpendiculars but instead of 90 degrees as the angle you can specify any value

In traditional systems of drawing,



Room Plan View

hatching is often used to signify that objects are of a solid nature There are many different types of hatching required for drawing conventions, so rather than supplying the DEGAS/GEM type fills, GFA have included a manual hatching system. Hatching can be done to any area with the actual lines taken from the stock drawing lines The slope of hatching from 1 to 90 degrees is one parameter and separation of line from 0 3mm to 4.5mm is the other parameter. A further function allows the same process to overlay in the opposite direction to create cross hatching

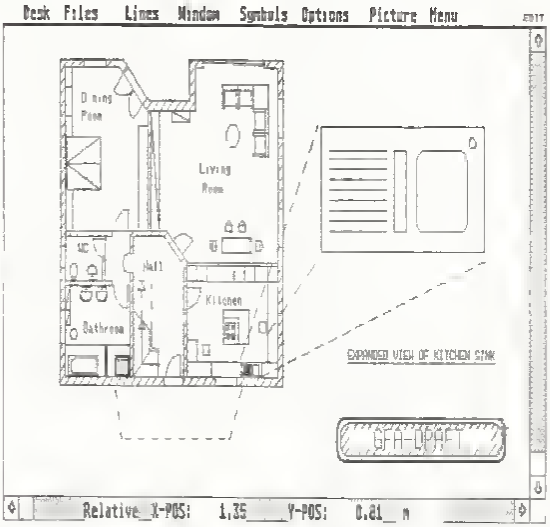
Construction lines are an important part of any scaled drawing especially objects with rounded corners These definable short tabs project out from your drawing to which your dimension lines and measurements are fixed The dimension line option is a separate function which interacts with the construction line facility By clicking the mouse on both construction lines a dimension line including the actual measurement is placed along the length of your object Line ends can be either arrowhead or oblique line, or if you wish can be left cut off The measurement can be displayed up to three decimal places

The final menu has the picture options Setting the scale is so easy from 1 1 to 1 1000 For scales up to 1 10 the co ordinates are shown in metres, for scales 1 1 to 1 9 they are displayed in millimetres Onca the scale has been selected all drawings remain encapsulated in that scale The scale can be in either inches or millimetres

There are four functions which relate to the picture itself deleting, moving, rotating and zooming, these are not to be confused with the 'window options Looking at these in a little more detail, the moving is achieved by typing in positive or negative values for the X and Y axis Rotating the picture is done in 90 degree anticlockwise directions The zoom factor can be from 0 01 to 9 99 in other words the picture can be reduced to one tenth of its original size or enlarged to ten times the original If the zoom factor is too large to enable the picture to fit in the drawing page the zoom does not take place The delete function operates as standard

In order to help placing objects correctly in the picture, a grid can be overlaid in preset spacings of 2, 4, 5, 8, 10 and 16 Unfortunately there is no way to define your own and relating it to the measuring system

Each picture that is being drawn is held in memory up to ten pictures (levels) can be held at any one time and as each level can be superimposed, a full picture can be built up from different screens As an example if you decided to show the assembly procedure for a piece of Flat-Pak furniture each level could be drawn separately and then superimposed to produca a finished picture If you linked this to printing or plotting to paper



Expanded View

as well as each level being printed individually they can be produced as one single drawing Further more, if you have a plotter each level can be plotted in a different colour subject of course to having the correct pens to level ratio

As you work on your picture the position of the cursor is displayed at the base of the drawing screen To enable accurate positioning in scaled drawings it's essential to measure the length of the lines exactly, this function has been well looked after by GFA There are three types of co ordinates display; absolute, relative and distance The default setting is absolute the co-ordinates are shown with reference to the origin at the bottom left hand corner, i e X=0, Y=0 Relative co ordinates are set at zero from an arbitrary point by clicking the mouse where you want to start your line and all your measurements are shown relative to the start position This is the best option for rectangles, which obviously have X and Y co ordinates (length and breadth) The final co ordinate is distance, in this option only one co ordinate is displayed, such as the length of a line

The second bar has five menus, the first two take on the printer and plotter routines respectively The default configurations are for Epson (doesn't say which type but presumably FX/LX) and Hewlett Packard 7475 for the plotter The first option is paper size, no preset sizes here so you have to rely on entering your own measurements

As well as the option for rotating windows and pictures in the drawing mode you can also rotate the picture in the printer/plotter function. This can only be done in one 90 degree rotation In order to position the picture in the optimum place on the paper an offset can be specified on both X and Y axis This takes the bottom left hand corner as its origin

Just as lines can be drawn and displayed in actual thicknesses or standard visual thickness the same applies to the printing Not to be outdone by the zooming for pictures and windows the printer/plotter routines can also decrease and increase the picture from a quarter size to a 4 times enlargement

The print density has an amazing seven levels, ranging from draft mode (single pass) to quadruple (triple pass) The quality of density level seven has to be seen to be believed. Curves are smoothed out superbly and look really natural If your picture is larger than the paper, GFA will let you insert fresh sheets of paper The plotter routines do not have a density option but makes up for that by plotting multilayered pictures in different colours

The preset menu has a number of toggled options, examples are cursor cross to complete screen crosswire As you move the cursor via the keyboard the stepping values can also be selected from as little as a pixel Text seems to be poorly represented with only four choices on disk and only two held in memory The text is one size only but can be increased or decreased by selecting a different zoom factor Although this sounds far from satisfactory It can be reasonably acceptable but only just' Perhaps to compensate text can also be rotated in 90 degree positions plus 'mirrored' if desired

GFA has one of the most unusual methods of erasure I've ever seen in a graphic program, the backspace and delete keys are used extensively throughout, fine for word processing but weird for drafting The undo key actually restores the last deletion Large scale deletion can be achieved by dragging a window over the object and deleting its contents There are a couple of extras to the delete functions which allow either/or text or lines to be deleted from the window This can be especially handy if the object area is difficult to get a window around it and you don't want to delete all of it.

The status manu just gives you the current situation of text choices, levels, scale symbol function keys and the amount of free memory The final menu lets you exit the program with or without saving the picture.

There are a number of points I personally didn't like, mainly cosmetic but important enough for me Most of them revolve around the inadequate manual, the lack of index and the constant jumping from command to command without any logical procedure such as following the menu There is a tutorial included in the manual and whilst it helps you through the mire it can be rather vague in places I found I had to interpret the instructions which was rather time consuming and very frustrating I found the constant switching from preset to definable operations disconcerting It looks as though two people wrote the program and neither of them would change their style!! There was a small problem over the printing of the tutorial picture which manifested itself when I tried to print it out at a specific reduction

I'm assured by Glentop that most of these points will be put right in the next version which should be out by the time you read this

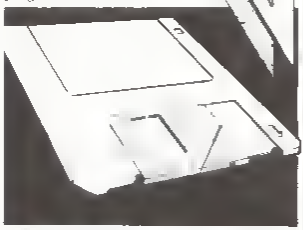
GFA DRAFT is a program which actually delivers the goods and undoubtedly will carve itself a nice position in the drafting class with its precise functions and many excellent facilities GFA DRAFT will draw attention from those who have outgrown the 'arty' programs and want to move on to a product capable of producing professional results

SOFTWARE FOR THE ATARI ST

ST Doctor

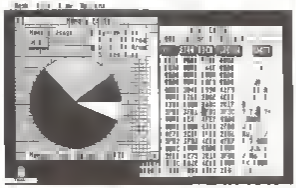
ST DISC

The utility disc for the Atari ST.
ST-Doctor is a Window-based desk accessory, available at any time whilst using other application programs



Main features include

- Complete scrolling window-based memory editor allowing any area of ST memory to be examined in ASCII, Hex, Decimal, Binary or Disassembly listing. Associated features include Copy memory block, Print block, Save block, Fill block, Search, and a pie chart showing memory usage



- Disc Editor: Similar editor to above, plus load/save file or sector, Advanced formatting, Search disc, Recover deleted file, show file size, show free disc space, etc
- Key recorder: A unique 'function key definer' this allows up to ten sequences of key-strokes to be assigned to any keys.
- Supplied with fully illustrated instruction manual

REVIEW

ST Doctor has to be described as one of the best if not THE best low level programmer's utility.

ST WORLD

Price £19.95 incl VAT

VISA

Access/Barclaycard

All orders despatched within 24 hours

Back-Pack

ST CARTRIDGE

A cartridge-based pack of general purpose utilities for every ST owner. When plugged into the cartridge port of any ST, all Back-Pack facilities are available whilst using GEM compatible applications and, because it resides in ROM, it occupies no valuable RAM space.

Scientific calculator
More comprehensive than most real calculators, this offers all the usual features such as Sin, Cos, Tan, Log, π, $X^{\wedge}Y$, $X\sqrt{Y}$, $1/X$, a memory store, etc. However, it also has programmer's features such as operation in Hex, Binary and Octal, plus logical operations. The result can easily be transferred into the current application

Price: £49.00 Incl

Clock and alarms
Analogue/digital clock showing time and date. Two time zones, four separate repeatable alarm settings, etc

Diary
A comprehensive diary allowing categorised entries on any day of the year. Simple controls make it easy to move forward or backward one day, month, or year.

Note Pad
Thirty selectable pages of any information can be stored in the note pad with cursor controlled editing and the ability to transfer information back to the current application or accessory in use

Typewriter
A simple utility allowing direct output to a printer at any time

Printer Buffer
Uses memory to 'queue' output for the printer, leaving the computer ready to use, whilst Back-Pack feeds the printer when it is ready

Address book
A computerised database of names, addresses and telephone numbers, allowing fast 'search' facilities to locate any entry

Ramdisc
Uses an area of memory, of specified size, to act as an extremely fast disc drive. A startup disc is easily configured to copy any list of files into the Ramdisc automatically

FAST ASM

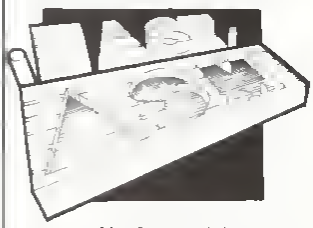
ST DISC

Fast ASM is an ultra fast assembler, it provides a user friendly environment for both writing and assembling 68000 programs.

REVIEW

It is an excellent assembler and at £19.95, sensational value for money too. I wouldn't SWAP it for anything'

ST WORLD



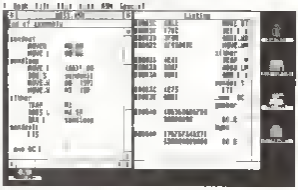
Main features include:

- Assembles up to 50000 lines per minute
- Program files are compatible with Fast Basic
- Able to run assembled programs without leaving assembler
- Provides fully relocatable PRG files
- Allows up to 10 source files in memory at one time
- Powerful GEM-based editor with mouse controlled cut, copy and paste facilities

REVIEW

Fast ASM is a very pleasant assembler to use and at 50,000 lines per minute, it certainly is fast. The environment is good thanks to an outstanding editor

ST WORLD



- Detailed source and object code listings
- Includes full instruction manual

Price £19.95 incl VAT

Computer Concepts

Gaddesden Place, Hemel Hempstead Herts. HP2 6EX 0442 63937

Fast ASM

From Computer Concepts
Price £19 95
Reviewed by Keith Mayhew

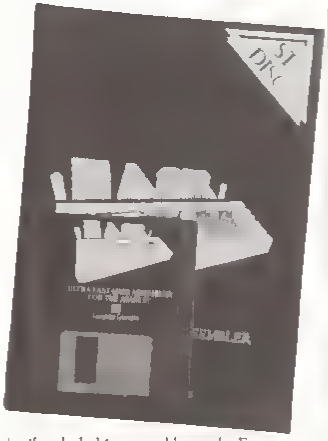
At just £19 95 Fast ASM is a well priced and very fast assembler for the ST Its assembly speed is stated at 50,000 lines per minute.

The assembler operates in a GEM environment with a powerful editor In fact, Fast ASM is derived from that very well known product Fast BASIC and so its similarity in appearance and use is hardly surprising

Source listings can be in upper and lower case although mnemonics have to be written in upper case When saved to disk, the listing is converted to tokens, thus saving spaca Alternatively, pure ASCII files can be read or written allowing some degree of interchange with other assemblers

Up to ten source files can be held in memory at once with the segmentation feature of Fast ASM making it easy to cut and paste between files The 'include' directive allows source files to be 'chained' together As each file is included it occupies a segment until it has been assembled This means that under the best conditions at least two files will be in memory at the same time If you intend to assemble very large programs you must ensure that any two source segments fit into your available memory in one go and allow room for assembly to progress It would have been nice to have had an 'assemble from disk' option that would remove this limitation

It appears that due to its close connection with Fast BASIC this assembler suffers a few defects Firstly, a few BASIC keywords can cause problems



if included in assembler code. For example, trying to use a label called 'MODE' is a sure recipe for disaster, the assembler crashes! Fortunately this seems to be an exception as most combinations of keywords work fine A problem, which is a design fault, is that labels can be redefined at any place in a source listing and, worse still, no warning is provided This could easily cause hours of debugging hell! This 'feature' is documented in the manual so the authors did know of its existence

Fast ASM will be a good purchase for a beginner and will probably serve the needs of most assembly language programmers Its major downfall is that it provides no 'advanced' facilities such as macros, which are almost a necessity for very large program development

M-Cache

From Microdeal
Price £24 95
Reviewed by Keith Mayhew

If you own a hard disk and yet you are still not satisfied with its performance then M Cache will answer your demanding needs

The program keeps a cache of hard disk sectors in memory so that access to these sectors is as fast as a RAMdisk. M-Cache allows you to specify how many sectors you want buffered, 200 is the recommended number for most purposes although it will require 100K. If you have a 512K machine you might consider getting a memory upgrade, but even a very small cache will help as long as there is enough to buffer directory sectors the worst offenders!

Supplied as a very small AUTO program, M Cache will provide you with the advantages of a RAMdisk while accessing your hard disk Its operation is totally transparent, all you see is the improvement in speed There is no need to worry about your data either, as sectors are automatically written back to disk as they are changed

Get this program if you are power hungry and demand the most out of your ST!

Tempus Text Editor

From Eidersoft
Price £34 95
Reviewed by Keith Mayhew

Described as a 'Turbo Text Editor' and 'The Professional Text Editor', Tempus has quite a reputation to live up to!

As far as speed is concerned, there is no doubt that this editor really is a turbo model, in fact, it appears to be rocket powered The reason for this is that the editor represents a whole year of assembly language programming, as is explained in the manual The professional part of the description is arguable, but this editor is definitely not short on features and represents one of the most comprehensive GEM based editors on the market

The manual supplied consisted of approximately eighty pages of detailed descriptions of the editor's features Although the manual was comprehensive and readable there were places where the English didn't quite make sense We have been informed that this will be corrected, as will a few other minor errors, before it goes on sale

The Tempus program is a little more than 55K bytes long, including all of its resource file information After loading, the editor will take all of the available memory barring 30K bytes Should desk accessories need more than this amount of free space then the editor can be configured to leave an appropriate size of memory free

The first thing that Tempus does is display its file selector box like 1st Word However, this is no ordinary file selector The user can select from one of ten predefined file extenders, e g *.*, *.C, *.DOC, or for one which is not provided for, can type it explicitly Changing drive is done at the click of a button and full disk and file information is available!

When a file is loaded, its name is assigned to an icon and a window is opened for it Up to four texts can be loaded at one time, but when a window is closed the text is still in memory and can be redisplayed by double clicking on its icon The advantage of this method is that you do not have to have your display cluttered with windows

The other icons on the desktop allow you to perform operations on texts by dragging one icon onto another For example a text can be erased by dragging it to the trash can and can be restored by reversing the operation You do not have to use the icons for all operations as they have been duplicated

WS REVIEWS ST REVIEWS ST REVIEWS ST

ROUTINES MATTER

By Marvey Mills Part 2

In the last issue we saw how an assembly code programmer may save a lot of time and avoid wasting memory space by using subroutines for program functions that are used many times For this program we will add one more routine to our subroutine library and also delve further into some of the more useful parts of the ST's operating system This program is a very simple one to display DEGAS pictures on screen. Very useful for a user who can't afford to splash out on DEGAS itself but still wants to take advantage of the many pictures around in the Public Domain or elsewhere

Getting a Line of Text

We saw in the last issue how easy it was to get a key from the keyboard for examination within the program (perhaps for a menu) This is all well and good if all you want to do is get one key, but what if you want a longer piece of text? You could put a routine together to repeatedly get single keypresses and add them together until RETURN is pressed, but this is very messy and GEMDOS already provides us with a means to perform this useful function The function is called 'READLINE' (surprise, surprise!) and is called in exactly the same way as our other subroutines The ST has its own keyboard buffer but we don't need to touch it at all All we have to do is tell GEMDOS where our text buffer is and then call READLINE The ST will then get all keypresses and put them into our buffer until RETURN is pressed Easy eh? Well if you think for a moment about this there are a couple of potential problems Probably the most important problem is that a buffer is a fixed length field in memory You can make it as big as you like but it still has a finite length Now then, what if the person doing the typing enters more characters than you have allocated to the buffer? Would GEMDOS continue to stuff charecters into memory, possibly overwriting program code? Suppose we don't reach the end of the buffer and hit RETURN within a safe limit, how can the program find out how many characters have been entered into the buffer? This could be important. Well both of these problems would be very serious if the authors of the operating system had been silly enough not to foresee them The solution is very neat and simple and totally overcomes both problems You first define the length of your text buffer by reserving a fixed area of memory Then, in the first byte of the buffer itself you write the total amount of characters that may be entered into that buffer This, of course, limits us to a maximum line length of 255 characters, but this is quite sufficient for most needs When you execute the READLINE function GEMDOS keeps track of how many characters it has written into the buffer and when you hit RETURN it writes the amount into the second byte of the buffer (it also puts this information into the D0 register) The actual text entered starts at the third byte of the buffer Something to be remembered is that you must take into account the fact that the first two bytes of the buffer are used by GEMDOS, so if you wanted a maximum line length of, say, 10 characters you have to reserve 12 bytes for the buffer If you forget this then GEMDOS will still write 10 characters to the buffer but the last two will overwrite whatever comes after the buffer space, with potentially disastrous results The function number of READLINE is $0A, and because it is a GEMDOS function it is a trap #1 call The only parameter you have to push onto the stack is the address of your text buffer in fact it is very similar to the print line routine discussed in the last article only it works in reverse! Its simplicity makes it ideal to make into a subroutine so that is just what we will do and then we'll add it to our subroutine library to stay in the basic 'framework' Since we are only using one area for the text buffer, there is no real need to pass the buffer address to the subroutine itself but if you were using multiple buffers you could pass the address of the one you want to use to the subroutine via the A0 register just like we did for the print line subroutine. All we have to do to call the routine is JSR to it When it returns to the program the buffer will have been filled and the length will have been written to the second byte of the buffer See Listing 1 for the subroutine

Just as printing to the screen conforms to the VT 52 standard in that ESCape characters have a special function, so does getting input from the user Listing 2 contains a list of the ESC codes you can enter instead of pure text and what they actually do You will see that CTRL-C not only ends text input but also aborts the program This can be very dangerous in certain cases but to get around this problem you would need to write your own input routine based around the GETKEY subroutine we discussed in the last article and also at the beginning of this one. This is quite a complicated task and CTRL C can be used quite safely here anyway so we'll stick with GEMDOS this time! Also, if CTRL J, CTRL M or the ENTER key are used to end text input then the final EOL

```
getline
                move.l    #ibuff,a0  ;get address of line buffer
                move.b    #20,(a0)   ;set 20 character max  input
                move.l    a0,-(sp)   ;push buffer address on stack
                move.w    #$0a,-(sp) ;push READLINE function number
                trap      #1         ;call GEMDOS
                add.l     #6,sp      ;correct stack
                rts                  ;return to program
```

Listing 1

```
CTRL-C    End text input and abort program.
CTRL-H    Backspace one character.
CTRL-I    Tab.
CTRL-J    Space line and end text input.
CTRL-M    Carriage return and end text input.
CTRL-R    Print line on new line.
CTRL-U    Start new line.
CTRL-X    Clear line and put cursor to start
```

Listing 2

(End of-line) character is NOT written to the buffer We will need this EOL character to be present later on in the program so always use RETURN when using this program.

Reading a Disk File

Now that we've got some of the easier uses of the operating system out of the way we can move on to the more complicated functions Nearly all programs will need to access a disk file at one time or another In this program we only need to read the DEGAS picture file We don't need to write files or perform any of the many other functions available for use when dealing with disk files, so we'll keep it simple There are three stages to reading a file successfully First you need to OPEN the file, this would have to be done whether you were reading or writing to the file Then you perform the actual READ itself transferring the data in the file to an area of memory Then when you have finished with the file you have to CLOSE it.

The operating system allows a program to have up to 40 files open at any time So in order to tell which open file you want to read or write to GEMDOS gives each a file 'handle' when you open it The handle is a special number that to GEMDOS means a particular file So if you opened a file and it was given the handle '2' then in order to perform an action on that file at a later time you would only need to tell GEMDOS to access file '2' and it would use the right one To open our DEGAS file we need to tell GEMDOS what the filename is We will already have the filename somewhere in RAM in our text buffer from when we used the GETLINE subroutine. This is where it is important that the EOL charecter is in the buffer as well. All we do is tell GEMDOS where the filename begins, it then reads it in from the buffer until it encountars an EOL character, so if it was missing the filename could be very long indeed as GEMDOS searches through memory! The only other parameter that the OPEN function needs is the 'file attribute This is a value that is saved in the disk directory against the filename and tells the operating system something about the file The only value we are interested in is '0' This is the attribute for 'normal file - read and write allowed' but if you want to know what the others are then have a look at Listing 3 Once these parameters have been passed and GEMDOS called then the operating system will go away and if all goes well, it will open the file and return to the program with the file handle (a word value) in D0 If all does not go well, it can't find the file on the disk or there is a disk error then it will return an error code in D0. The error code will always be negative so it is easy to find out if something went wrong by simply testing for a minus number If you want to know what the specific problem was then you have to know what each error code actually means For this program we will not be going into much detail, we will only need to know that an error actually occurred, but Listing 4 contains the list of GEMDOS error codes for when you want your error trapping to be more helpful Note that they are not all to do with disk I/O (input/output), some are used with memory allocation and reservation Once you have been given a valid handle for the opened file you must store it safely in memory You need to use it every time you want to access the file and OPEN is the only time you call the file by its actual filename

Once the file has been opened it is ready for you to read from it. The READ function needs three parameters passed to it First of all you need to tell GEMDOS where in memory it is going to store the information that it reads from the disk file Then it needs to know how many bytes to read from the file and also the handle of the file that you wish to read from When you have pushed this information onto the stack and called GEMDOS the data will be read into memory at the address you gave it. When it has finished, GEMDOS will return to the program with the number of bytes read successfully in the D0 register If an error occurred during the read the D0 will contain the error code Again, we are not concerned with the type of error, only that one did, in fact, occur

When you have finished with a file you must close it This is to inform GEMDOS that you no longer need access to that file and also to make the handle available for reuse All you need to do to close a file is push the handle number onto the stack and call another GEMDOS function Listing 5 shows you the general procedure for opening, reading and closing a file

Colour by Numbers

In order to display the DEGAS file we will need to do two reads, this is because DEGAS saves its picture information in two chunks The first chunk which is 34 bytes long is the picture header information (data saved at the start of the file containing information relevant to the picture itself). This is where the palette data is stored and we need to read this information to a special buffer in order to use it. The palette data is made up of 16 words of information Each word corresponds to a colour The ST can display 16 colours in low resolution, 4 in medium resolution and 2 in high resolution. but DEGAS saves all the colour palette incetions regardless of the screen resolution used to create the picture. It can detect which resolution was used by examining the first word in the picture header This contains a value that corresponds to the correct resolution according to the following table

1 = Low resolution
2 = Medium resolution.
3 = High resolution

We will not worry about the correct resolution, if you simply ensure that the current resolution is the same as that used in creating the picture all will be well All we have to do to install a new palette is get the address of the palette data by adding two to the address of the buffer used to read in the data, and call

```
$00  =   Normal file, read/write enabled.
$01  =   File is read only.
$02  =   File is hidden i.e does not appear in directory
$04  =   System file.
$08  =   File is volume label i.e disk name.
$10  =   File is a subdirectory.
$20  =   File is written and closed correctly.
```

Listing 3.

```
-32  Invalid function number
-33  File not found
-34  Pathname not found
-35  Too many open files (no handles left)
-36  Access not possible
-37  Invalid handle number
-39  Not enough memory
-40  Invalid memory block address
-46  Invalid drivespec
-49  No more files (used when searching directories)
```

Listing 4.

an operating system function The SETPALETTE function is an XBIOS call (as opposed to GEMDOS which we have been using so far) and is thus a trap #14 call These calls perform in exactly the same way as the GEMDOS calls. The only difference is that they use a different part of the operating system See Listing 6 for the method of installing a new palette

Touching Base

Once we have read in, and installed our new palette we need to read in the 32K of picture information Where do we store it? Well, we could set up a buffer 32K long and read it into that, but there is no need DEGAS saves its picture data by simply copying the screen RAM to disk If we just copy it back again then, hey presto, we are displaying the picture exactly as it would be displayed by DEGAS Screen RAM is the area of memory that holds the bytes that go to make up the screen image so all we need to do is find out where the screen RAM begins in memory How do we do this? Easy, another XBIOS function does it for us. The ST is the easiest computer I've ever had to program, the operating system does EVERYTHING for you (well almost!) There are actually two screen RAMs in the ST, the Logical screen base and the Physical screen base which are usually set to point to the same area of memory The Physical screen base (physbase) is the address that the display chip is actually using to start displaying data from, and the Logical screen base (logbase) which can be different from physbase, is set to a supposed screenful of data upon which all screen operations like print line and clear screen are performed You could, for example, set physbase to point to a picture while logbase was set to another area of RAM in which you were setting up a menu Then when you had finished setting up your 'virtual' screen you would set physbase equal to logbase and instantly flip from picture to menu Since physbase is usually set to the same area as logbase we can use the logbase value as the buffer for our data This means that when we call GEMDOS to read the picture data, it will read it straight into the screen RAM area See Listing 7 for the LOGBASE function

```
open
        move.w  #$0,-(sp)       ;set file attribute
        move.l  #name,-(sp)     ;push address of filename
        move.w  #$3d,-(sp)      ;push OPEN function number
        trap    #1              ;call GEMDOS
        add.l   #8,sp           ;correct stack
        tst.w   d0              ;test D0 for errors
        bmi     error           ;if minus then branch to error handler
        move.w  d0,handle       ;save file handle

read
        move.l  #buff,-(sp)     ;push buffer address for read data
        move.l  #34,-(sp)       ;set to read 34 bytes
        move.w  handle,-(sp)    ;push file handle
        move.w  #$3f,-(sp)      ;push READ function number
        trap    #1              ;call GEMDOS
        add.l   #12,sp          ;correct stack
        tst.l   d0              ;test D0 for errors - note that you
                                ;must use longword test in this case
        bmi     error           ;if minus then branch to error handler

close
        move.w  handle,-(sp)    ;push file handle
        move.w  #$3e,-(sp)      ;push CLOSE function number
        trap    #1              ;call GEMDOS
        add.l   #4,sp           ;correct stack
        tst.w   d0              ;test for errors
        bmi     error           ;branch if error
```

Listing 5.

```
        move.l  #pal,-(sp)      ;push address of palette data
        move.w  #6,-(sp)        ;push SETPALETTE function number
        trap    #14             ;call XBIOS
        add.l   #6,sp           ;correct stack
```

Listing 6.

```
        move.w  #3,-(sp)        ;push LOGBASE function number
        trap    #14             ;call XBIOS
        add.l   #2,sp           ;correct stack
```

Listing 7.

About the Program

If you take a look at the program itself (Listing 8) you will see that it is mostly divided up into blocks, each of which is a subroutine or function that we have talked about so far The very first instruction after the memory management header is actually a Line A call that disables the mouse display This is a particularly useful function for TOS programs because otherwise the mouse can be moved around leaving gaping blobs of green background in the middle of the screen Hardly the 'Leonardo De Vinci' art we have come to expect from DEGAS! Then we clear the screen and print the program title and prompt. The prompt asks the user to input the name of the file to be displayed which we get by using the GETLINE subroutine Straight away we check the part of the buffer that contains the number of characters entered If it is zero then the user just hits return without entering a name In this case we quit the program and return to the desktop

When a valid name has been entered then the program attempts to open the file If, for some reason, it cannot do this then we jump to an all-purpose error handler that informs the user that an error occurred and then restarts the program If the file opens OK then the file header is read in and the new palette is implemented Once this is done then it gets the logical screenbase address and reads the file into screen RAM Again if there is an error during the read then we use the all-purpose error handler

Once the picture is being displayed then the program just sits and waits for the user to hit a key We could have used the GETKEY subroutine in the subroutine library for this but there is a problem with that routine in that is also prints the character entered to the screen which disturbs the picture What we need to use is another GEMDOS call that gets a key without printing it to the screen Function #7 ofGEMDOS, called 'Direct CONIN (console input) without Echo' does just this When the user hits a key then the program restarts so that another picture can be displayed

```
;*****************************************
;* DISPLAY - A DEGAS picture display utility
;* demonstrating some more routines and
;* operating system calls.
;* By Harvey Mills for Monitor magazine
;*****************************************

lf      equ     $0a             ;set up some constants
cr      equ     $0d             ;for the print formatting
eol     equ     $0              ;characters - wakes program
esc     equ     27              ;more readable
nosnow  equ     $a00a           ;A-Line call - hide mouse cursor

        ;** Memory Management Header **

        move.l  a7,a5           ;get old stack pointer
        move.l  #ourstack,a7    ;and set stack pointer to our area

        move.l  4(a5),a5        ;add up all the various parts
        move.l  $c(a5),d0       ;of memory that our program
        add.l   $14(a5),d0      ;uses and tell GEM to keep them
        add.l   $1c(a5),d0      ;safe for our program
        add.l   #$100,d0        ;otherwise GEM could overwrite the
        move.l  d0,-(sp)        ;program during its
        move.l  a5,-(sp)        ;operation
        move.w  #0,-(sp)
        move.w  #$4a,-(sp)
        trap    #1
        add.l   #12,sp

;*********************************
;       PROGRAM CODE HERE       *
;*********************************

start
        dc.w    #nosnow         ;hide the mouse
        jsr     clrscrn         ;then clear the screen

        move.l  #title,a0       ;get address of title string
        jsr     pline           ;and call our print line routine

ask
        move.l  #prompt,a0      ;get address of the prompt
        jsr     pline           ;and print it

        jsr     getline         ;get filename to display

        cmp.b   #0,ibuff+1      ;were characters entered?
        beq     quit            ;go on quit program

        ;** open disk file. **

        move.w  #0,-(sp)        ;file attribute
        move.l  #ibuff+2,-(sp)  ;push address of filename in ibuff
        move.w  #$3d,-(sp)      ;push OPEN function number
        trap    #1              ;call GEMDOS
        add.l   #8,sp           ;correct stack
        tst.w   d0              ;error?
        bmi     derror          ;yes- process disk error
        move.w  d0,handle       ;opened Ok - save file handle

        ;** read palette data **

        move.l  #pichead,-(sp)  ;push address of header buffer
        move.l  #34,-(sp)       ;push number of bytes to read
        move.w  handle,-(sp)    ;push file handle
        move.w  #$3f,-(sp)      ;push READ function number
        trap    #1              ;call GEMDOS
        add.l   #12,sp          ;correct stack
        tst.l   d0              ;error?
        bmi     derror          ;yes- process disk error

        ;** new new palette **

        move.l  #pichead+2,-(sp);push address of palette
        move.w  #6,-(sp)        ;push GETPALETTE function number
        trap    #14             ;call XBIOS
        add.l   #6,sp           ;correct stack


        ;** get screen address **

        move.w  #3,-(sp)        ;push LOGBASE function number
        trap    #14             ;call XBIOS
        add.l   #2,sp           ;correct stack
                                ;d0 now contains screen address

        ;** read picture data **

        move.l  d0,-(sp)        ;push address of screen ram
        move.l  #$8000,-(sp)    ;32k to read
        move.w  handle,-(sp)    ;push file handle
        move.w  #$3f,-(sp)      ;push READ function number
        trap    #1              ;call GEMDOS
        add.l   #12,sp          ;correct stack
        tst.l   d0              ;errors?
        bmi     derror          ;yes- process disk error

        ;** get a keypress **

        move.w  #$07,-(sp)      ;push RCdE function number
        trap    #1              ;call GEMDOS
        add.l   #2,sp           ;correct stack
        jmp     start           ;jump back to start of program



;*****************************************************************
;* START OF SUBROUTINE LIBRARY                                   *
;*****************************************************************
clrscrn
        move.l  #scrncode,a0    ;get address of ESC code for clear
        jsr     pline           ;call our print line routine
        rts                     ;return to program
scrncode
        dc.b    esc,'E',eol,eol ;escape code for clear screen
        even

quit
        move.w  #0,-(sp)        ;push function number on stack
        trap    #1              ;call BOOTOS
                                ;good thats it- does not return

pline
        move.l  a0,-(sp)        ;get address to start print
        move.w  #$09,-(sp)      ;push the function number on stack
        trap    #1              ;call GEMDOS
        add.l   #6,sp           ;correct stack
        rts                     ;return to program

getkey
        move.w  #$01,-(sp)      ;push function number on stack
        trap    #1              ;call GEMDOS
        add.l   #2,sp           ;correct stack

        and.b   #%11011111,d0   ;include this line if you want to force
                                ;the character into uppercase

        rts                     ;return to program with character
                                ;code and scancode in register d0
```

```
getline
        move.l  #ibuff,a0       ;get address of line buffer
        move.b  #28,(a0)        ;initialize max size
        move.l  a0,-(sp)        ;and push address on stack
        move.w  #$0a,-(sp)      ;push readline function number
        trap    #1              ;call GEMDOS
        add.l   #6,sp           ;correct stack
        rts                     ;and return to program

derror
        move.l  #errmsg,a0      ;get address of error message
        jsr     pline           ;call print line routine
        jsr     getkey          ;get a key
        jmp     start           ;jump to program start

;************************
;* Text storage
;************************


title
        dc.b    ********************************** ,cr,lf
        dc.b    *                                * ,cr,lf
        dc.b    *            DISPLAY             * ,cr,lf
        dc.b    *                                * ,cr,lf
        dc.b    *   DEGAS Picture display utility * ,cr,lf
        dc.b    *     By Harvey Mills for Monitor * ,cr,lf
        dc.b    ********************************** ,cr,lf
        dc.b    lf,lf,eol

prmpt
        dc.b    cr,lf
        dc.b    " Enter filename to display) ,eol

errmsg
        dc.b    cr,lf
        dc.b    " A disk error has occurred. ,cr,lf,lf
        dc.b    " Press any key to re-run... ,eol


;************************
;* data storage
;************************
                even
                ds.l    64      ;user stack area
newstack        ds.l    1
saveosp         ds.l    1       ;save supervisor stack pointer
ibuff           ds.b    22      ;buffer for getline routine
picthead        ds.w    17      ;picture header buffer
handle          ds.w    1       ;save file handle

        end
```

Listing 8.

ANIMATIC
Atari ST Image Editor plus
For colour systems with Fast Basic
AniMatic is a great new animation design package
With it you can easily produce full colour movable animated objects for use in your own Fast Basic programs
Use it to create animated graphics for games Produce your own computerised cartoons
Let your imagination run wild!
AniMatic is compatible with NEOchrome, so you can combine animation with NEOchrome backgrounds. Or use the NEOGRAB function to cut sections from a NEOchrome screen, animate them and overlay them onto the original screen in your own Fast Basic program
An Matic objects can be any size up to 30 pixels square. They can be loaded into your own Fast Basic programs moved and animated under program control.
The AniMatic disk comes with several demonstration programs including the game Smashout. Together with six program modules written in Fast Basic containing set up and manipulation routines which can be inserted directly into your Fast Basic programs.
At this price AniMatic must be the best value software of its type for the ST. Try it, we are sure you won't be disappointed.
Please note AniMatic is specifically for users of Computer Concepts FAST BASIC It is not directly compatible with other versions of Basic.
only £9.95 including P & P
Please send cheque or P.O. to –
SOFT BITS Dept. M
5 LANGLEY STREET
LONDON WC2H 9JA
TEL 01 836 2533
SOFT BITS

ATARI 400/800/XL/XE UTILITIES

Check out our NEW Summer/Autumn Prices!

GAME PROGRAMMERS! Our **PROGRAMMERS UTILITY PACK** comprises of a set of game development utilities suitable for you whether you program in **BASIC** or **ASSEMBLER** and whether you program for **PROFIT** or just for **FUN**
The P.U.P. will enable you to: Design and animate single and multicoloured Players: Create static and vertically scrolling game Playfields: Define custom multicoloured character sets: Create musical data for background music and sound effects. Includes many additional helpful routines. Supplied with full documentation.
TWO DOUBLE SIDED DISKS AND INSTRUCTION MANUAL. £22.95 Post Free (plus FREE GAME).

DISK-TO-DISK UTILITY. Our very popular disk backup system features: a **SECTOR COPIER** to copy fully **128 sectors** groups of sectors or entire single or double density disks. A **DISK MAPPER** to locate Used and Unused sectors plus all **BAD SECTORS**. A **FAST FORMATTER** suitable for bulk formatting. A **BAD SECTOR CREATOR** to enable you to create **Bad Sectors** and backup CRC error **Bad Sector** protected disks. Very easy to use. **NEW LOW PRICE. DISK £9.95 Post Free.**

TAPE TO-DISK UTILITY. Automatically transfers all your commercial single stage machine code games from cassette to **DISK.** Load your favourite games in a fraction of the time with 100% reliability. **NEW LOW PRICE. DISK £8.95 Post Free.**

MENULOADER. If used with our Tape to Disk utility you may place **TEN** transferred game tapes onto **ONE** disk. The games may then be loaded with a single keypress. (Menuloader will copy itself.) **NEW LOW PRICE. DISK £7.95 Post Free.**

THE SECTOR INSPECTOR. A Disk Scanner and Sector Editor. Adventure disks may be searched for clues and locations, and any existing text within a game can be replaced with your own amusing or risque comments. Great fun! **NEW LOW PRICE. DISK £8.95 Post Free.**

TURBOCHARGER XL/XE. Bored with those lengthy cassette loading times? Turbocharger will backup your tapes **AND** reduce their loading times! **NEW LOW PRICE. TAPE £7.95 Post Free.**

BINARY TAPE EDITOR. Offers tape backup **PLUS** the option to scan your tapes for text. If you get stuck on an adventure simply scan the tape for clues. **NEW LOW PRICE. TAPE £7.95 Post Free.**

TAPE TO-TAPE UTILITY. Similar to Turbocharger but without the Turbo feature. The Tape to-Tape utility will backup single and multistage, long and short IRG tapes. **NEW LOW PRICE. TAPE £6.95 Post Free.**

THE CASSETTE DUPLICATOR. A tape backup utility written in Basic. The program may be LISTed. **NEW LOW PRICE TAPE £6.95 Post Free.**

THE ALCHEMIST. In addition to all of your singlestage tapes, the Alchemist will also transfer many of your multistage tapes to **DISK.** The utility can transfer many of these cassettes only' files to disk. **S.A.E. or Phone for details.**

FREE GAME! We will supply 'Exterminator' an original machine code action game **FREE** with all orders over £20. All software is supplied complete with easy to follow instructions. Following day despatch of orders and free postage (overseas customers should add £2 to order). S.A.E. or Phone for further details.

Cheques and P.O.'s to

J.R. DAVIS (Dept M)
10 INGRAM AVENUE, HOLMER,
HEREFORD. HR4 9RD. TELE: (0432) 50836

ST PROGRAMMING

By Keith Mayhew BSc(hon) Part Three

Bindings

Access to an operating system routine from a language such as C is achieved through a simple function call For example 'appl_init(), is the standard call to initialise an application program with the AES We shall now consider how these functions are implemented This information will not be essential reading for a C programmer but it will provide a useful knowledge should you ever need to implement or change a function in any way; perhaps to get a faster function call If you intend to code in assembly language then the following is essentiall

Virtually all compilers sold for the ST have several libraries of functions which provide access to the operating system routines via standardised names These libraries are called bindings because they provide an interface between a language and the machine code calls to the operating system thus binding them together

As any assembler programmer will know a call to the operating system is always made via a 'TRAP' instruction Of the sixteen traps available on the 68000, TRAP #0 through TRAP #15, only four are used by the ST

TRAP #1 — GEMDOS
TRAP #2 — VDI and AES
TRAP #13 — BIOS
TRAP #14 — XBIOS

We shall look at the calls to the BIOS, XBIOS and GEMDOS in a future article You should find these functions well documented and fairly easy to follow They also all share the same simple calling sequence. A function number is pushed to the stack followed by any parameters and the TRAP call is made. Afterwards the stack is adjusted to remove the numbers which were pushed, for further details refer to Marvey Mills 'Routines Matter' in issue 16

Calls to the VDI and AES are very much more complicated than any other TRAP call which is often why assembler programmers don't like working with GEM! The bindings for say C hide all these complications behind small functions such as 'appl_init() '

Calling the VDI

All communication with the VDI routines is performed via five arrays

Array/Purpose
contrl[12] / Pass control info
intin[128] / Pass integers in
ptsin[128] / Pass points in
intout[128] / Pass integers out
ptsout[128] / Pass points out

All of these arrays consist of 16 bit words Information is passed to the VDI via the arrays contrl, intin and ptsin, information is passed back via contrl, intout and ptsout Note that contrl is used in both directions, that is the VDI can change values in the array on return

Each of the arrays must be large enough for any of the functions you call Also note that the size of an array usually depends on how much data you wish to pass For example, text is passed to the VDI via intin and so if you only print small strings you only need a small array Remember though that the arrays must be large enough for the worst cases and you must always have intout and ptsout large enough to receive information back from the VDI The sizes specified above are typical values which will suit most applications if you are concerned about memory usage you could always reduce the array sizes if possible

Rather than pass the address of each of the five arrays separately to the VDI, a further array is used, called a parameter block. The size of this array is fixed at 20 bytes consisting of five 32 bit pointers to the five arrays in the order in which they were listed above.

A call to the VDI consists of loading register D0 with $73, loading D1 with the address of the parameter block array and performing a TRAP # 2 Note that the value in D0 specifies that a VDI call is being made Figure 1 illustrates the array set up and a possible calling sequence in assembler

When a VDI function is called from C, the code which forms the function, or binding copies data into the input arrays and initialises the control array with information, such as the opcode number, identifying the function to be performed, the handle of a workstation, and the lengths of the input arrays After calling the VDI the binding copies any return data back to the calling program

The 'points' arrays are used wherever co ordinate positions are specified Each point is stored as two 16 bit words in the array, the first giving the X value and the second giving the Y value Note that an array of 128 words will therefore only hold a maximum of 64 points The 'integer' arrays are used to pass any other type of information such as line widths, colour, etc

The binding functions supplied with most C compilers for the VDI, allocate the space for the parameter block array but do not reserve any space for the other five arrays. The library code for the VDI bindings access these arrays as the external names of contrl, intin ptsin, intout and ptsout

The advantage of this method is that the space allocated for the arrays is not predetermined, but is defined by the programmer Every program which is to be linked to the VDI must therefore define these arrays as global data If you get the names wrong then the linker will inform you that some of the names have not been defined but are referenced by the VDI code

Calling the AES

A call to the AES is very similar to the VDI register D0 is loaded with $C8, D1 is loaded with the address of a parameter block and a TRAP # 2 is made The array set up for the AES and a possible calling sequence is illustrated in Figure 2

First of all, it is important to note that all of the arrays for the AES are totally separate from the VDI's, including the parameter blocks

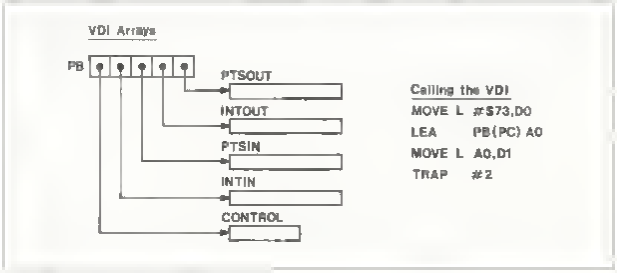


Figure 1

The arrays 'ADDR_IN' and 'ADDR_OUT' are used for passing the addresses, or pointers to data structures. The arrays 'INT_IN' 'INT_OUT' and 'CONTROL' have similar purposes to their counterparts used by the VDI. The 'GLOBAL' array is used by the AES to store information about the application and should generally not be altered by the application directly. An 'appl_init' call initialises the 'GLOBAL' array and its contents are used by all subsequent calls to the AES.

The parameter block is 24 bytes long and contains six pointers to the other arrays. The 'CONTROL' array consists of 5 words: the opcode of the function to be called and the sizes of the integer and address arrays in terms of the number of entries they can hold.

Common VDI Calls

Due to the vast number of VDI functions I have decided not to cover them all in great detail, at least for now. The following will provide a very broad overview of the functions available and explains a few of the most common functions used in programs. Once you get a feel for the VDI functions you should find it very easy to follow most documentation available. If you have any particular difficulties with some functions then they will probably become the subject of a future article.

The VDI library of functions can be catagorised by their purpose. They are as follows: Control, Output, Attribute, Raster, Input, Inquire and Escape.

Control Functions

Control functions deal with the broad aspects of workstations such as opening and closing them or changing fonts. Last time, we discussed workstations and the use of the v_opnvwk and v_clsvwk functions for opening and closing them. The handle number returned by the v_opnvwk call has to be remembered as it is needed by virtually all calls to the VDI. You can decide either to keep your handle global or whether to pass the handle to all your functions which access the VDI. The second method has the advantage of being flexible in that if you have more than one workstation open then you can use the same function on each of them.

The other control function which is essential to virtually all programs is 'vs_clip(handle, flag, &points[0]);'. If flag is zero then all output to the workstation identified by 'handle' will, in subsequent calls, not be clipped, i.e. it will draw whatever you ask it to. If flag is non zero then the array 'points' specifies the upper left and lower right co-ordinates of a clipping rectangle as four words. Note that it is the standard in calls to the VDI that all points are stored as absolute co-ordinate pairs with the X co-ordinate stored before the Y.

Once a clipping rectangle has been specified then all further calls will only produce output in that area. It will be seen that clipping is essential when working with AES windows. The only disadvantage of clipping is that ALL output is slowed down, but not necessarily by the same factor, usually the process will be imperceptible, but under a few circumstances you might pay large speed penalties. However, it is often possible to avoid speed reductions by using a few simple tricks which we will cover in the future.

Output Functions

The three output functions which we will cover here allow us to draw lines, fill rectangles and print text. 'v_pline(handle, count, &points[0]),' is used to draw any number of lines in one go. The array 'points' contains a list of points with 'count' specifying how many there are. 'v_pline' then draws lines between the first and second points, the second and third points, etc , until the last two points have been joined. If only two points are specified then a single line will be drawn. To plot a single point you should use two points set to the same co-ordinates, i.e. a short line. Do not use a single co-ordinate as it will not be displayed. 'vr_recfl(handle, &points[0]) ' fills the rectangle specified by the two points in the array. It is useful for clearing areas inside windows, etc.

'v_gtext(handle, x, y, &text[0]),' prints the null terminated string of characters (bytes) in the text array at the point specified by x and y (in pixels).

Attribute Functions

The attribute functions determine how the output functions draw their images on the workstation.

The most important function is 'vswr_mode(handle, mode),' which determines the writing mode, i.e. how images are drawn over a background image. It affects the operation of just about every output function so ensure it is properly set before you use any particular function.

If 'mode' has the value of 1 then the new image replaces the old, thus printing text would cause both its background and its foreground colours to be plotted.

Transparent mode is set when 'mode' is 2, this causes only the foreground colour of an image to be drawn. So printing text in this mode would draw only the text and would not clear the rest of the character's cell: this is useful when text needs to be superimposed over a graph for example.

XOR mode is set with a value of 3 for 'mode'. If an existing pixel is of background colour then the new pixel will be drawn as in the replace mode. However, if an existing pixel is in foreground colour then the pixel to be drawn will be inverted, i.e. if it was to be in background colour then it will be plotted in foreground colour and vice versa. Note that this mode gives the ability to draw an image once and then to draw exactly the same image again to undo or erase it.

Lastly a value of 4 in 'mode' sets the drawing mode to reverse transparent. This is the same as the transparent mode except that the image to be drawn is reversed so that its foreground colour is where its background colour was and vice versa. Thus if you print text in this mode the character will be displayed in inverse. However, remember that the character itself, i.e. the background colour will not be plotted, so to guarantee a proper inverse character it should be drawn on a rectangle of background colour.

The rest of the attribute functions are broadly split into groups which correspond to groups of output functions, for example, there are a set of attribute functions for text which set colour, size, rotation, effects, etc. There is another set which determine the colour, thickness, style, etc of line drawing. And so on.

The function 'vsf_interior(handle, style),' is used to determine how areas are filled. A value of zero for 'style' produces a fill in the background colour whereas a value of one will fill with the foreground colour, other values allow for patterns or user defined styles. Use of the 'vr_recfl' function with one of the above settings will provide a way of clearing areas of the screen, as for example when a window has to be redrawn.

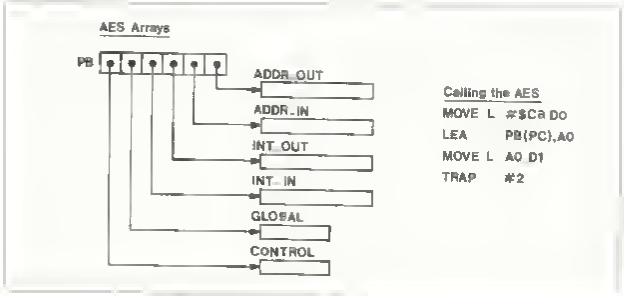


Figure 2

Raster Functions

There are a few functions available in the VDI for performing raster, or blitting, operations Their use is mainly for moving rectangular bit images around the screen and are often used for scrolling, as it is faster than having to redraw the entire image

Input, Inquire and Escape Functions

These groups of functions form quite a large part of the VDI, in terms of the number of functions yet are probably the least used, although there are a few essential functions among them!

The Input functions provide methods of sampling the keyboard and mouse etc As most programs use the AES they tend to use the functions provided by the AES for these devices You always have plenty of choice, for example, you can read the keyboard from the BIOS, GEMDOS, VDI or AES!!

The Inquire functions provide some useful information about the current configuration of hardware and software as to which parts of the VDI are supported and to what levels Much of the real use of these functions is in providing truly portable code which checks a system's capabilities before calling on certain functions This type of feature will not interest most ST users as they know what their machine does! However, there are a few functions which provide quite valuable information which you cannot obtain elsewhere in the system so they are certainly not redundant

Lastly escape functions implement a host of miscellaneous functions which tend not to fall into any other category

Next Time

Although we will study the VDI in detail in the future, hopefully this has given enough of an insight into its basic organisation and uses

Next time, we will start a very in depth examination of the AES, looking at its principles and showing that the initial shock is really not as bad as it might seem

Please note that in part two, Listing 2 was printed the wrong way round in error It was NOT a late April fool!!!

CLASSIFIED

The classified section is for private individuals only (not companies) to buy and sell computer hardware, software, make contacts, find pen pals, etc All adverts will be free up to 30 words, thereafter the charge will be 10p per word (cheques and postal orders made payable to the club) Send your advert to us by the 1st December for the next issue together with any payment necessary Please mark your envelope 'Classified'

For sale 800XL, 1050 disk drive with lazer, 1010 tape deck touch tablet, joystick some tapes and disks £170 May split. Phone Robert on (0706) 354577 after 4pm

COMPUTE!'s Books Of Atari, 1 2 3, Machine Language for Beginners, £5 each Wanted Mapping the Atari Book Two Phone 021 328 6853 or write to K. Berry, 50 Brantley Road, Birmingham B6 7DR.

Pen Pals wanted I have 800XL and 1050 disk drive and would like to hear from other eight bit users Write to M. Hutton, 18 Riverside Road, Ipswich IP1 4AR

Wanted Atari magazines for new Atarian Cassettes (originals) £1 each or swap Contact Mrs Hamlyn, 37 Sedgeley Grove, Elson, Gosport, Hants PO12 4NL

Adventures. Deadline, Enchanter and Sorcerer £12 each or £30 for all three disks Send SAE for a complete list. D Purvis, 6 Llanbleddian Gardens, Cardiff CF2 4AT

Games Assorted tape and cartridge, send for list Offers accepted, all originals **Brand New** Bell & Howell Super 8mm sound movie camera, autofocus, electric zoom, etc. Cost £250, used twice, swop 1029 or 1050 (must be working and in good condition) Offers considered **Spectrum 48K,** some extra books, £40 ono Delivery by arrangement or postage at cost Ring Syd, Crawley 511708 after 4pm, (Sussex area)

ST Contacts. I wish to make contact with ST computer owners Write to Grahame Farrall, Shipton House, Shipton Under Wychwood, Oxon OX7 6BA

800XL for sale. Almost new, plus 1010 cassette recorder and 2 games £60 Phone 01 778 6033 after 4.30pm Ask for Wayne

For sale Maplin speech synthesiser £33 Star Raiders cartridge £6 50 Datasoft Basic compilar £12 Spectrum 48K computer, interface, joystick and software £73 Diskey utility £12 Zork II £10 Mask of the Sun £16 Wishbringer £12 Wizard and the Princess £8 Blade of Blackpoole £8 Tel 0502 66026

Fast Basic User Group: First issue of club newsletter available soon and will cater for those beginning and the more advanced user. Many new and bright ideas including a beginners section to teach you to program Good programmers and authors wanted to help run the newsletter Please send a SAE for further details to FaST, c/o Simon Rush, 42 York Road, Rayleigh, Essex SS6 8SB

T W A U G The Tyne and Wear Atari User Group welcomes new members, if you would like to know more write to F Gregory, 20 Mapla Ave, Silksworth, Sunderland, Tyne & Wear SR3 1DW

The Gateway Club Is a multi computer club with an Atari section which holds meetings once a month For more info contact Bob Hall on (0638) 717723 or write to 46 Churchill Drive, Mildenhall, Suffolk IP28 7DA

Bargain! Atari operating system Source Listing, as new still in wrapper, 400/800 version but suitable for 800XL owners Only £10 including p&p Ring (0702) 617307 after 6pm.

For Sale. Small business system Atari 130XE, dual 1050's, Philips green monitor and Epson FX80 printer Atariwriter Plus, Syncalc, Synfile, Bookkeeper, Disk Utilities and games £500 Phone Fareham (0329) 239036

COLOUR CHART

By Trevor Skeggs

Want a colour chart to adjust your TV with? This simple program uses static player/missiles to produce 5 broad colour bands.

```
100 READ X: IF X
255 THEN M=X: GOTO 100
110 TRAP 120: POKE M,X: M=M+1
GOTO 100
120 DATA 53248,38,70,102,134,166,174,
182,190,3,3,3,3,255,255,255,255
130 DATA 255,623,17,704,36,166,
14,0,0,0 8 0
```

Press Shift plus Clear to clear the screen of text, then move the cursor to any black area. The first four numbers after 53248 represent the positions of the four bars and may be adjusted accordingly (e g by adding 4 to each) The next four numbers position the fifth bar The four numbers following 704 determine the first four colours used, and the penultimate number is the fifth Keep the last and third from last numbers the seme, to avoid seeing the text window area as a separate colour

BACK ISSUES

Previous issues of Monitor are obtainable from the club for £1 plus 30p postage each. They contain many interesting and informative articles hints and tips, program listings, reviews and practical advice If you have missed out send for your copies of back issues today!! Please note that issues 1,2,3 4,5,6 & 7 are already sold out

Number 8

Includes Cracking the Code 2 new series, Opening Out and Starting from Basics. Horizontal and vertical scrolling Mask of the Sun, Sorcerer Conan Alley Cat Ghostbusters and Spy vs Spy all reviewed Programs include Quickplot, Nightmare Reflections and Matchbox

Number 9

Includes. RAMdisk for the 130XE Appraisal of the 130XE Introduction to MIDI Keyo typing checker Binary loads from Basic. Reviews of TopDOS Homeword and Mr DO! Overview of FORTH Programs include Fast shape filler, Multiboot index and Happy Typer

Number 10

Includes. All about digitised pictures How disk files work Cracking the Code Starting from Basics and What s MIDI all continue. Programs include Disk Jacket, PCB Paranoia and 3D Maze American Road Race, Kennedy Approach, Asylum, Red Moon and Wishbringer reviewed

Number 11

Includes RAM Talker for 400/800 Book reviews MIDI programs ST Hi-res Hat program Hexadecimal Code generator Reviews of Atariwriter Plus, Sidewinder, Koronis Rift, Electraglide Mercenary, Fighter Pilot, Goonies and Alternate Reality Plus Starting from Basics and Cracking the Code.

Number 12

Includes Add-on circuits for various motors. Disk file handling Metrices and Arrays explained Write your own adventure Space Invaders program Reviews of Technicolour Dream, Eidolon and Action Biker ST reviews include DB Master One, Time Bandit and Menu Plus

Number 13

Includes Omnimon and Ultimon compared Data compression Megamax C and Lattice C evaluated Tamper the sound of your 8 bit Players and missiles explained. Programs include Graphics 8 page flipper, Demon adventure game Reviews of Super 3D Plotter II, Planetarium, Price of Magik, Last V8 and Nuclear Nick ST reviews include Cornerman Cards and Major Motion.

Number 14

Includes Display Lists. Adventurers sentence analyser In depth look at Happy Revision 7 Graphics Modes Video digitiser mods for use with XL/XE machines Deathzone, a superb arcade game Reviews of Crystal Raider, Molecule Man, Domain of the Undead, Lasar Hawk, Rick Hanson, Colleen Music Compendium and Spellbreaker ST reviews include Music Studio Starglider, TrimBasa, Electronic Pool, Easy Record and Pinball Factory

Number 15

Includes Player/missile priorities and Interrupts Turbo Basic commands and functions 1050 write switch project. Enter commands directly in Basic Whist card game for you to type in DOS modifications OS Controller Card evaluated Reviews of Splitfire 40 Crumbles Crisis Robot Knights and Replay Intro to ST programming ST Blitter Reviews of Hollywood Hijinx, BCPL K Resource Make Micro time Clock Card Alternative, Trivia Challenge and Fast Basic.

Number 16

Includes Character mapped modes and an introduction to scrolling Using PLOT and DRAWTO in Graphics Zero A useful hexadecimal converter program. Minotaur a machine code monitor from Basic Split screen effects for adventure writers XIO for beginners Mini Office II Autoduel Death Race Sprong and Space Lobsters reviewed ST section includes How to use GEM with examples in C Useful routines written in assembler Six ST books reviewed Hades Nebula, Airball, ST Replay ST Digidrum Crafton & Xunk, Animatic, Zoomracks II Mousetrap, Prohibition and Barbarian are all reviewed

MONITOR ON DISK

All the 8 bit programs published in each issue of Monitor are now available pre recorded on disk for you No more need to spend frustrating hours of typing only to find the program won t run, and you are faced with the daunting task of bug hunting The price is just £4 95 which includes postage and packing. Send a cheque/postal order made payable to the 'U K Atari Computer Owners Club' to Monitor Magazine P O Box 3, Reyleigh Essex SS6 8LR. If you live in Europe add 50p If outside Europe add £1 00 Please allow 14 days for delivery

Monitor Disk 8

Includes. Berg, an entertaining adventure mystery set on the high seas Smooth vertical and horizontal scrolling demos.

Scrabble crossword game Bonus program a sector editor written in Action!

Monitor Disk 9

Monitor Disk 10

Monitor Disk 11

Monitor Disk 12

Monitor Disk 13

Monitor Disk 14

Monitor Disk 15

Monitor Disk 16

Monitor Disk 17

SUBSCRIPTION FORM

If you are not already a subscriber fill in this form (or a photocopy) and send it to the address below together with a cheque/postal order made payable to the 'U K Atari Computer Owners Club'. Your subscription entitles you to receive the next four issues of Monitor and enrols you as a member of the club. Please state from which issue number your subscription should commence. Annual subscription rates are £5 00 in the U.K and Eire, £8 00 in Europe and surface delivery outside Europe, £12 00 Airmail delivery outside Europe

Don't delay do it today!!

I want to enrol as a club member and receive Monitor magazine I enclose a cheque/postal order for £5 00/£8 00/£12 00 Please send me issue ____

Name _____

Address _____

Post Code _____

My system is an XL/XE ☐ ST ☐ XL/XE and ST ☐

Send to Monitor, P.O. Box 3, Reyleigh, Essex SS6 8LR.

ST ST

StockSoft

520STFM ..... lowest prices in the UK.
1040STFM ..... for latest prices see our price list

Public Domain Software: Games, Utilities, Art, Demo's, and Nudes.
Wide variety, all at only £5 per disk, post free.

ST HAPPY

Happy computers of the USA who have been making superb disk backup utilities for the Atari 8 bit range since 1979, have now announced an ST backup cartridge which will back up 3.5" standard ST disks, also 3", 3.5", 5.25", and 8" disks form any other computer, provided the compatible disk drive is connected.
For further information apply below.

BLANK DISKS

High quality GoldStar branded disks, tested and certified 100% error free. Single sided, box of 10, only £15, post free.
Double sided, box of 10, only £20, post free.

For further information on any of the above please send large S.A.E. to:-

STOCKSOFT

15 WOODBROOKE ROAD, BIRMINGHAM, B30 1UE.

MAIL ORDER ONLY. ATARI ONLY.

NEW LOW PRICE ST!

ONLY FROM SILICA

Finally, there's a personal computer that not only solves problems like other computers but also solves it's own problem! The other computers have created. At Silica Shop we pressed to present the ST range of purpose-built business computers from Atari. The ST was designed utilising the most advanced thought in semiconductor technology to produce a computer that gives the user very low power costs. Which means it costs less to make. And less to buy. The latest ST computers now include built-in power supplies and built-in disk drives. The TOS that controls them and GEM which controls your screen are now on ROM chips which are already resident in the ST keyboard. They operate automatically from the moment you switch on. Since Shop are pleased to offer the complete Atari ST range. Our mail order department is situated in Sidcup, and we have 2 retail outlets at Sidcup, Lion House (Tottenham Court Road tube station) & Oxford Street. We have eight years experience of Atari products, longer than any other UK company, and are well established as the UK's No 1 Atari specialist. With a group turnover of over £4 million and in excess of 50 staff, we offer you unbeatable service and support. We provide several showrooms which you will find invaluable during your Atari computing life and most of these facilities are available ONLY FROM SILICA. We suggest that you read through what we have to offer before you decide where to purchase your Atari ST.

FREE STARTER KIT - Only From Silica

When you purchase any Atari ST keyboard, you will not only receive the best value for money computer on the market, but you will also receive the following from Atari Corporation as part of the package:

BASIC Language Disk *BASIC Manual* *ST Owners Manual* *TOS/GEM on ROM*

If you buy your ST from Silica Shop, you will also receive:

1st Word *Word Processor*
ABC/Database Database - your graphics program

In addition, we at Silica would like to see you get off to a flying start with your new computer so we have put together a special **ST STARTER KIT** worth over £100 which we are giving away **FREE OF CHARGE** with every ST computer purchased at our normal retail prices. This kit is available **ONLY FROM SILICA** and is aimed at providing users with a valuable introduction to the world of computing. We are additionally upgrading the ST Starter Kit, which contains public domain and other licensed software as well as books, magazines and accessories all relevant to ST computing. Return the coupon below for full details.

DEDICATED SERVICING - Only From Silica

At Silica Shop we have a dedicated service department of seven full time Atari trained technical staff. This team is totally dedicated to servicing Atari computer products. Their accumulated knowledge, built up over the years, makes them second to none in their field. You can be sure that any work carried out by them is of the highest standard. A standard of servicing which we believe you will find **ONLY FROM SILICA**. In addition, we are pleased to offer a collection facility for Atari ST computers (both in and out of warranty) so there is also no need to remove any of the memory expansions to ST computers.

1Mb RAM UPGRADE: Our upgrade on the standard Atari 520ST-M or 520ST-FM keyboards will increase the memory from 512K to a massive 1Mbyte (1 Meg). It has a full 1 year warranty and is available from Silica at an additional retail price of only £99.95 (+VAT = £115).

TV MODULATOR UPGRADE: Silica can upgrade the 520ST-F to include a TV Modulator so that it can be used with your domestic TV set. Fitting is quick and simple and does not require any soldering or special tools. A cable to connect your ST to any domestic TV is included in the price of the upgrade which is only £49 (inc VAT). The upgrade is also available for early 520ST computers at the same price.

THE FULL STOCK RANGE - Only From Silica

We aim to keep stocks of all Atari related products and our warehouse carries a stock value in excess of £1 million. We support many software titles direct from the USA and you will find that we have new releases and a range of many of our competitors. Unlike dealers who only stock selected titles, we have the full range in addition we carry a complete line of all books which have been written about the Atari ST. A range as wide as ours is something you will find is available **ONLY FROM SILICA.**

AFTER SALES SUPPORT - Only From Silica

Most customers find that when you buy your ST from Silica Shop, you will be fully supported. Our free mailings give news of releases and developments. The newsletters are up to 48 pages long and are crammed full of information. We will be happy to answer any queries you may have by telephone or letter. Unlike dealers who may be here today and gone tomorrow, we have been established for over 8 years and are financially secure. Our technical support helps you get the most from your computer. Because we have both the staff and the systems specifically dedicated to providing after sales service on Atari ST computers, we are confident that our users enjoy an exceptionally high level of support. This can be received **ONLY FROM SILICA.**

FREE CATALOGUES - Only From Silica

At Silica Shop we recognise that serious users require an in-depth information service, which is why we produce two newsletters and a price list each month. These are up to 48 pages long and are crammed full of information. We will be happy to answer any queries you may have by telephone or letter. Unlike dealers who may be here today and gone tomorrow, we have been established for over 8 years and are financially secure. Our technical support helps you get the most from your computer. Because we have both the staff and the systems specifically dedicated to providing after sales service on Atari ST computers, we are confident that our users enjoy an exceptionally high level of support. This can be received **ONLY FROM SILICA.**

FREE OVERNIGHT DELIVERY - From Silica

Most orders are processed through our computer within 24 hours of receiving order. Most hardware orders are sent by the next night BROUP 4 courier service FREE OF CHARGE to customers within the UK. This method helps to ensure minimum delay and damage in transit.

PRICE MATCH - Only From Silica

We hope that the combination of our low prices, FREE UK delivery service, FREE Starter Kit and FREE after sales support will be enough to make you buy your Atari equipment from Silica Shop. If however you see a system offering you a similar package, and you find one of our competitors offering a lower price, please write to us with their name and address and the details of the offer. We will be happy to match any genuine offer on a like for like basis (we do not price match end of year sales, bankrupt offers, etc.) and will provide you with the same fast, free delivery. We feel sure that we are able to beat most competitors' prices. However, if you come to us for a price match and we cannot beat the offer submitted, we will still be happy to match it! Please telephone our Sales Department on 01-309 1111 and ask for the Price Match Manager. **ONLY FROM SILICA.** We do not price match other companies claiming to offer a price match system.

SIDCUP (& Mail Order) **01-309 1111**
1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX

LONDON **01-580 4839**
Lion House (1st floor), 227 Tottenham Court Rd, London, W1P 0HX

LONDON **01-629 1234 ext 3914**
Selfridges (1st floor), Oxford Street, London, W1A 1AB

£260 +VAT=£299

SO MUCH FOR SO LITTLE!

There is nothing that can compare with the incredible value for money offered by Atari's 520ST-FM. For only £260 (+VAT £299) you can purchase a powerful 512K RAM computer with a 95 key keyboard (including numeric keypad), MIDI interface, GEM, a palette of 512 colours, mouse controller and a 512K built-in disk drive. The 520ST-FM has a TV modulator built in and comes with a lead to allow you to plug it straight into any domestic colour television set. The mains transformer is also built in to the keyboard, so there are no messy external boxes. You couldn't wish for a more compact, powerful and stylish unit. Atari ST computers are now firmly established in the UK, there are nearly 500 software titles already available for a wide variety of applications and the list is growing all the time. And that's not all. When you buy your new 520ST-FM (or any Atari ST computer) from Silica Shop, you will get a lot more, including a FREE Silica ST Starter Kit worth over £100. Read the ONLY FROM SILICA section on the left, to see why you should buy your new high power, low price 520ST-FM from Silica Shop, the UK's No1 Atari Specialists. For further details of the range of Atari ST computers and the FREE Silica ST Starter Kit, complete and return the reply coupon below.

ATARI 520ST-FM NOW ONLY £260 (+VAT=£299)

520ST FM with 512K RAM & mono monitor £395 (inc VAT) — Upgrade from 512K RAM to 1024K RAM £100 (inc VAT)

ATARI 1040ST-F NEW PRICE

We are pleased to announce a new reduced price for the 1040ST-F which is now available for only £449 (inc VAT). The 1040ST-F is a powerful computer with 1Mbyte of RAM and a built-in 1Mbyte double sided 3½" disk drive. The 1040 has been designed for use on flat mains and professional applications, most of which require a high resolution monochrome display. It does not therefore have an RF modulator to allow it to be connected to a domestic TV set. Modulators can be fitted for £25 (inc VAT).

1040ST-F Keyboard — Without Monitor — £449 (inc VAT)
1040ST-F Keyboard — High Res SM125 Mono Monitor — £599 (inc VAT)

Do you want the further details of the 1040ST-F, return the coupon below.

MEGA STs NOW IN STOCK

Also on sale were megabyte even more RAM than the 520 or 1040. The new MEGA ST computers are now available with 2 or 4 megabytes of RAM internally. These are expandable to a massive 4Mb. Both new computers are fully compatible with existing STs and run currently available ST software. The MEGA STs are styled as an expandable Central Processing Unit with open architecture and a detachable keyboard. They come complete with GEM, a free mouse controller and all extras supplied with the 520 or 1040. Prices are as follows:

MEGA 1 Keyboard, CPU, Internal Double Monitor — £999 (inc VAT)
MEGA 2 Keyboard, CPU, Internal Double Monitor — £1199 (inc VAT)
MEGA 2 Keyboard, CPU, Internal Mono Monitor — £1099 (inc VAT)

Would you like details of the MEGA STs, return the coupon below.

ATARI ST

To: Silica Shop Ltd, Dept ATCOC 1087, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX

PLEASE SEND ME FREE LITERATURE ON THE ATARI ST

Mr/Mrs/Ms: Initials: Surname:

Address:

Postcode:

Do you already own a computer? If so, which one do you own?